JPRS-UEQ-89-008

2 JUNE 1989

FOREIGN BROADCAST INFORMATION SERVICE

JPRS Report

Science & Technology

USSR: Engineering & Equipment

JPRS-UEQ-89-008

2 JUNE 1989

SCIENCE & TECHNOLOGY

USSR: ENGINEERING & EQUIPMENT

CONTENTS

NUCLEAR ENERGY

Skill Training and Maintenance Systems for Power Generating Unit Operators
[A. F. Dyakov, R. D. Tsiptsyura; ELEKTRICHESKIYE STANTSII, No 3, Mar 89]............................... 1

Controlling the Parallel Operation of Synchronous and Asynchronized Turbogenerators
[V. P. Oleksin, A. I. Matviychuk, et al.; ELEKTRICHESKIYE STANTSII, No 3, Mar 89]............... 15

INDUSTRIAL TECHNOLOGY, PLANNING, PRODUCTIVITY

On Building a Heterogeneous Local Computer Network for Developing Flexible Manufacturing Systems
[N. P. Starodub, A. I. Slobodyanyuk, et al.; UPRAVLYAYUSHCHIYE SISTEMY I MASHINY, No 6, Nov-Dec 88]. 24

Language Tools for Planning the Control of Advanced Electronic Automatic Equipment
[V. L. Sosonkin, L. Ye. Shergin; UPRAVLYAYUSHCHIYE SISTEMY I MASHINY, No 6, Nov-Dec 88].................. 30

Automation of Decision-Making to Neutralize the Consequence of Unscheduled Situations in Flexible Manufacturing Systems
[N. V. Globa; UPRAVLYAYUSHCHIYE SISTEMY I MASHINY, No 6, Nov-Dec 88]....................................... 40

Industrial Computer Network - A Systems Engineering and Process Functioning Base of the Integrated Automated Control System
[A. A. Morozov, Z. M. Aselderov, et al.; UPRAVLYAYUSHCHIYE SISTEMY I MASHINY, No 6, Nov-Dec 88].. 48

Base Local Computer Network Complex Hardware
[Ye. P. Moiseyenko, G. I. Sinyayev; UPRAVLYAYUSHCHIYE SISTEMY I MASHINY, No 6, Nov-Dec 88]. 55

MISCELLANEOUS

Information Retrieval System of Archival Space and Aerial Remote Sensing Data
[Yu. V. Chukin; UPRAVLYAYUSHCHIYE SISTEMY I MASHINY, No 6, Nov-Dec 88]...................................... 60

Abstracts of Articles in MASHINOVEDENIYE, No 2, 1989
[MASHINOVEDENIYE, No 2, Mar-Apr 89].................... 64

UDC 621.311.25:621.039.007.2:658.386.001.5

Skill Training and Maintenance Systems for Power Generating Unit Operators

18610459A Moscow ELEKTRICHESKIYE STANTSII in Russian No 3, Mar 89 (signed to press 22 Feb 89) pp 13-20

[Article by Candidates of Technical Sciences A. F. Dyakov and R. D. Tsiptsyura, USSR Power and Electrification Ministry and Kiev Automation Institute]

[Text] Much attention has been focused in print media on what should happen to the skill training and maintenance system for power generating enterprise personnel, primarily of generating unit operators at fossil-fuel (TES) and nuclear electric power plants (AES). We should note that there is no unanimous opinion about the principles of developing and operating such systems. Various structure options of the system have been offered [1-4] and their advantages and shortcomings have been described. Recently, the opinion that staff practice and training systems and their individual elements affect generating unit performance have been expressed in a number of sources, e.g., [5]. The opinion that integrated practice and training systems encompassing various aspects of staff work must be designed is often expressed, since the use of simulators solely to develop operator skills and habits, regardless of basic theoretical knowledge, did not yield the desired result. An idea of carefully picking one, even the most important, skill whose accumulation affects the success of operator activity as a whole, and concentrating solely on developing this skill, has not produced a positive outcome either. In order to answer the question about the necessary operator skill training and maintenance system, we must first analyze operator activity.

Many studies by industrial psychology experts are centered on describing operator activity, e.g., [6, 7]. An attempt is made in a number of sources [8, 9] to describe the work of power generating unit operators. One more way to describe generating unit operator activity – graphically (which, in the authors' opinion, is more pictorial than a narrative) – is cited below. A diagram of operator activity is shown in Fig. 1, and Fig. 2 shows the structure of the relations among the operator, production equipment, process, control system, and data display system which are characteristic of generating unit operators. Let us briefly examine each diagram.

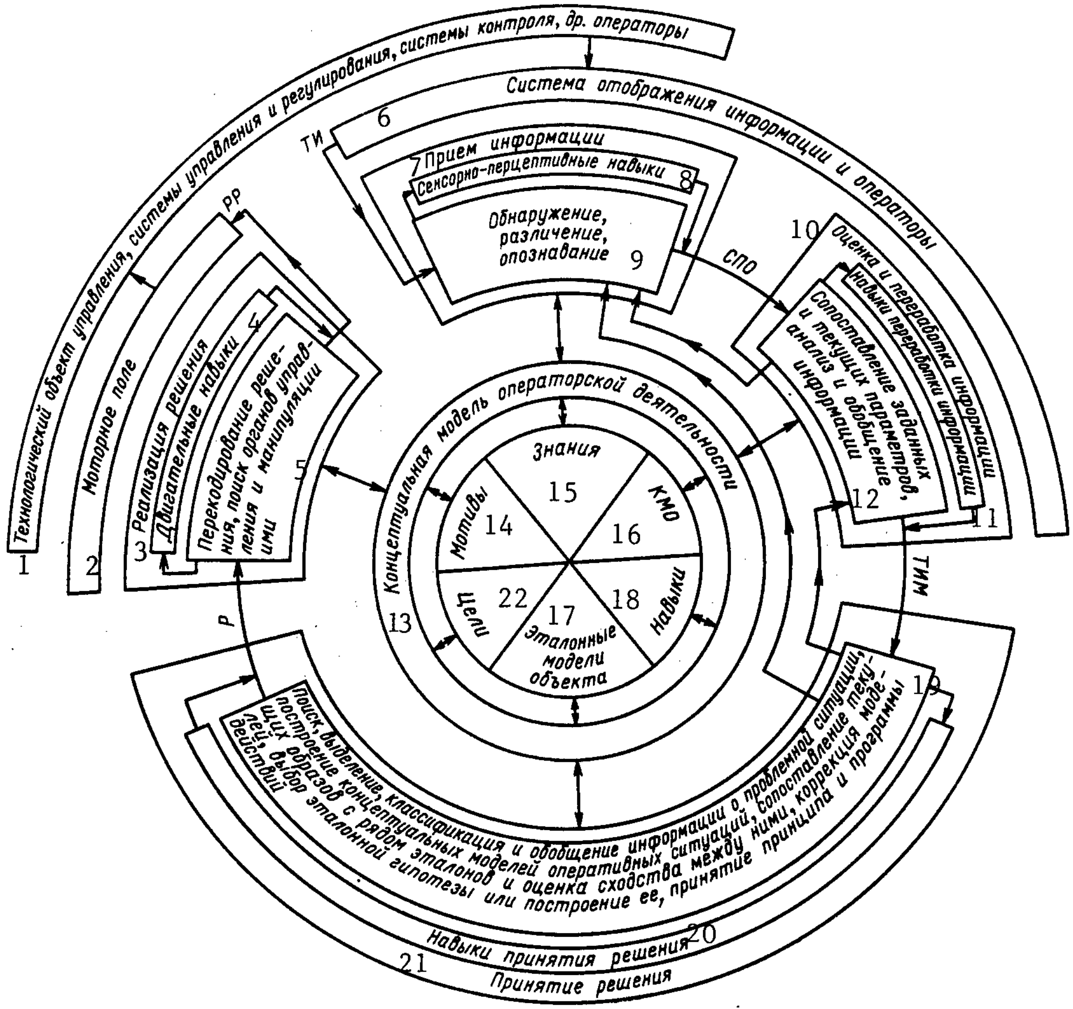


Fig. 1. Operator activity diagram: СПО (SPO) – sensory–perceptive image; ТИМ (TIM) – current information model; P – decision; PP – realizing a decision; ТИ – current information; КМО – conceptual model of the object; 1 – controlled plant, management or control systems, monitoring systems, and other operators; 2 – motor field; 3 – realizing a decision; 4 – motor skills; 5 – recoding a decision, finding and manipulating controls; 6 – data display system and operators; 7 – receiving data; 8 – sensory–perceptive skills; 9 – detection, discrimination, recognition; 10 – estimating and processing information; 11 – information processing skills; 12 – comparing set and current parameters, analyzing and generalizing information; 13 – conceptual operator activity model; 14 – motives; 15 – knowledge; 16 – КМО; 17 – benchmark plant models; 18 – skills; 19 – searching, identifying, classifying, and generalizing data on problem situations, constructing conceptual models of operational situations, comparing current images to a number of benchmarks and evaluating their similarity, correcting models, selecting a benchmark hypothesis or framing it, and adopting a principle or a program; 20 – decision-making skills; 21 – decision-making; 22 – goals.

The diagram's outer ring shows the four main activity phases which, essentially, compose any operator work: receiving information, evaluating and processing information, making a decision, and realizing the decision thus made. It is stated in [7] that human operator activity is a continuous, integral process; yet, despite this fact, it can be generally represented as consisting of the aforesaid four principal stages (under the same names as those used here). The same outer ring cell containing the names of activity phases list the types of skills necessary for executing each phase, and a brief description of the nature of work carried out to this effect.

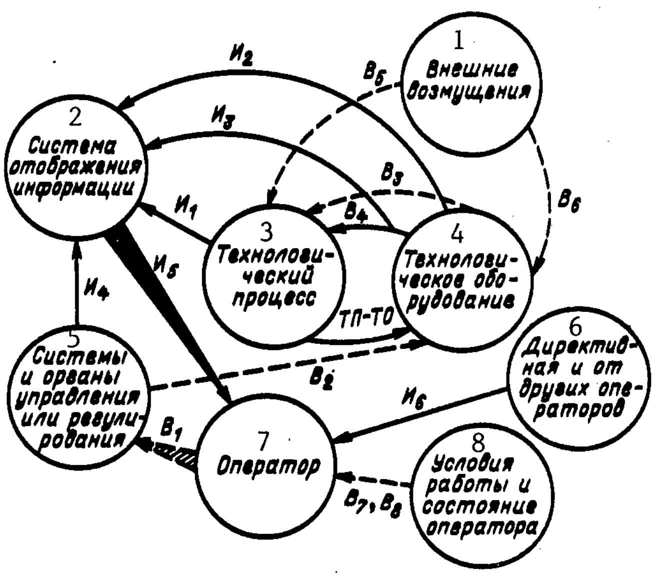


Fig. 2. Structure of relations in the work of a generating unit's ASU TP operator: И_1 – dataflow describing TP; И_2 – dataflow describing TO status; И_3 – dataflow describing TO–TP relation; И_4 – dataflow describing the status of control systems and controls; И_5 – data received by the operator; И_6 – directive data and data from other operators; B_1 – operator action on systems and controls; B_2 – action of control systems and controls on TO; B_3 – TO–TP control action; B_4 – TO–TP interaction; B_5 – effect of external perturbation on the TP course; B_6 – effect of external perturbation on TO status; B_7 – effect of operator's work conditions on his current activity; B_8 – influence of the operator's functional state; 1 – external perturbation; 2 – data display system; 3 – process; 4 – production equipment; 5 – control systems, controls, and actuators; 7 – operator; 8 – work conditions and operator state.

It follows from the chart that during the first phase of his activity, an operator receives current data (TI) from data display systems (SOI). The data reception stage includes detection, discrimination, and recognition skills. In engineering psychology, this class of skills is referred to as sensory–perceptive. During the second activity phase (evaluating and processing information), the operator uses the sensory–perceptive image formed as a result of his first phase activity as a source image. During the second phase, the operator compares the values of set (allowed) and current parameters and analyzes and generalizes the information. The outcome of his second phase activity is the current information model (TIM). During the third stage, the operator makes a routine decision based on the TIM formed in his mind, as well as available knowledge and accumulated experience and understanding of the conceptual model of the object's behavior, allowing for the goals and motives guiding him (the operator) at a given time moment or in a given situation. And finally, during the fourth phase, the operator realizes the decision he made by interacting with remote control, automatic control, and shielding and interlocking systems through terminals (keys, switches, buttons, etc.) located in the motor field of the control panel,

after which he checks variations in the TI as a whole and individual parameters which he has affected at the preceding stages.

The components which, when added up, form the so-called conceptual model of operator activity (KMOD) are located at the center of the diagram. In the authors' opinion, the KMOD can be classified as a conceptual model of the object (KMO). Each operator has an individual KMOD formed while mastering his profession, as well as practicing it, on the basis of a sum of theoretical and concrete knowledge and the results of analyzing the experience accumulated during his previous activity, allowing for the goals and motives guiding the operator in his practical work, based on his profound understanding of the so-called benchmark models of the object (the latter is the TIM's correspondence to the status of production equipment and process). According to the definition in [7], a KMO is a "mental picture" of the controlled process and conditions under which it occurs. In our opinion, the KMOD characterizes more fully operator activity and serves as a kind of theoretical foundation of virtually all operator actions, which is reflected in the diagram: the KMOD is positioned at its center, while the arrows indicate that the operator refers to KMOD at all the stages of his activity.

What conclusions can one draw from the above diagram? First, that operator activity is integral, continuous, and cyclical, and consists of a successive execution by the operator of certain phases. Second, that the cyclical nature of activity may, if necessary, be interrupted by returning to the preceding phase or even to the one before that. This is necessitated by the requirement to refine TI or the need to clarify the TI evaluation and processing results.

Today, the persons operating transport (vehicles, locomotives, ships, and aircraft) and processes and engineering systems (electric power plants, power systems, gas distribution systems, and airports) are called operators. A generating unit operator is a process operator. The principal relations among the operator, control and data display systems, production equipment, and the process are shown in Fig. 2. The work of a generating unit operator is characterized in that it is based on the information received from SOI. According to data compiled at the Engineering Psychology Department of the Leningrad State University, about 80% of erroneous operator actions at all phases of his activity is related either to a flawed perception of current data (various mistakes related to detection, recognition, and reading) or to an erroneous evaluation of the data perceived.

A generating unit operator perceives data primarily from SOI (temperature, pressure, or flow rate readings); consequently, one of the key issues is his ability to convert these data to understanding the production equipment status and grasping the course of the process, i.e., based on his knowledge and experience, the operator must learn to process such data into images characterizing the status of the process and production equipment.

Summing up our analysis of the diagrams in Figs. 1 and 2, we can draw the conclusions which are important for developing operator skill training and maintenance systems.

Conclusion One. Generating unit operator activity is a continuous process with recurring cycles. Thus, selecting one or several – even the most important – phases of activity and focusing all operator attention on them during training without regard for other phases is flawed. Only integrated training aimed at developing all the components of operator activity is fundamentally correct.

Conclusion Two. To operate successfully, an operator must take the following steps:

acquire enough knowledge for working independently in the generating unit's automated process control system (ASU TP) loop;

develop a range of skills for receiving, evaluating, and processing information, skills to make and carry out decisions, and skills to correlate data from SOI with the status of TO and the course of TP;

gain a profound understanding of the so–called benchmark models of the controlled entity status and benchmark models of the course of the process, as well as real–time images displayed by the SOI;

comprehend the interrelations and interdependence between the production equipment status and the course of the process (interaction B_4 in Fig. 2); and

understand the goals and motives which must guide him in his activity.

Conclusion Three. We have already indicated that erroneous operator actions are caused primarily by receiving, evaluating, and processing current data. Consequently, training must be carried out under the conditions which approximate reality as closely as possible, i.e., if data on a real power unit control panel (BShchU) is displayed on screen, the skill of receiving these data must, in the final analysis, be taught by reading data on screen. The same holds true in the case of data output on various instruments. The principle whereby the simulator BShchU data field is similar to the BShchU data field of a real ASU TP must be strictly adhered to. This principle does not rule out a psychological similarity of the simulator SOI to the real controlled object SOI. Yet, while refining skills and habits in order to develop a know–how for controlling a generating unit, the adequacy of the simulator BShchU data field to that of a specific BShchU is a must.

Conclusion Four. The ultimate goal of any system for raising the professional competence of generating unit operators and maintaining it at a high level is to guarantee a set level of service staff performance.

Operator skill training systems for generating unit operators in the power industry, both current and planned, are analyzed below from the viewpoint of the above assumptions.

Conventional Training System. Generating unit operator training organization and rules are based on the directives of the USSR Ministry of Power and Electrification [10, 11]. A

streamlined block diagram of such a system extensively used at TES and AES is shown in Fig. 3.

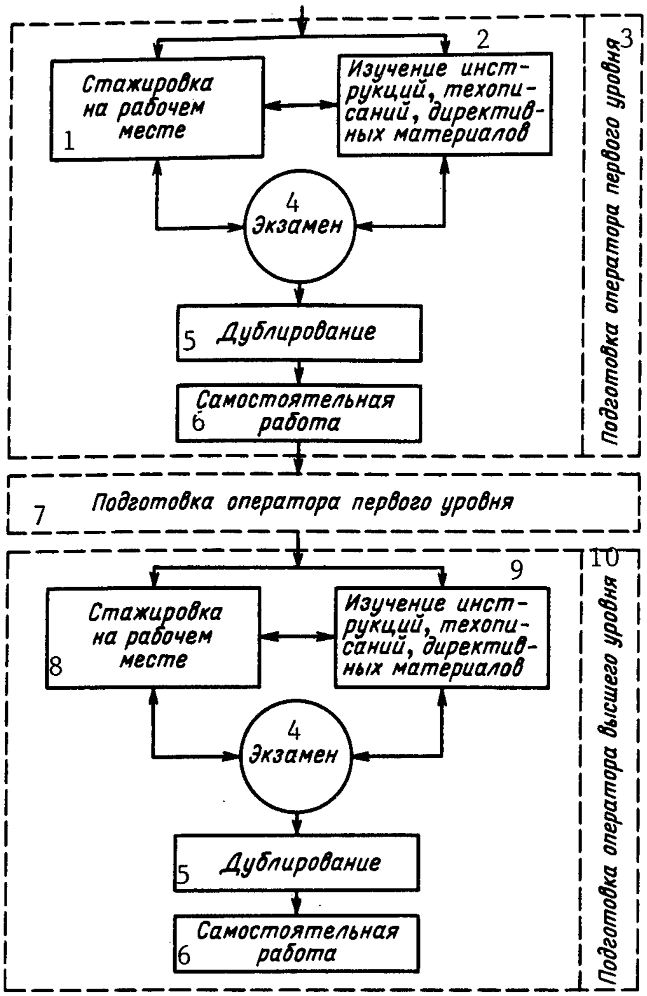


Fig. 3. Block diagram of conventional training and skill advancement for generating unit ASU TP operators: 1 – apprenticeship; 2 – studying manuals, directives, and specifications; 3 – entry–level operator training; 4 – test; 5 – duplication; 6 – homework; 7 – entry–level operator training; 8 – apprenticeship; 9 – studying directives, manuals, and specifications; 10 – top–level operator training.

Such a system is characterized by the following features. First, training as a whole is comprehensive since it combines theoretical (studying manuals, specifications, directives, and other documents) and practical (on–the–job apprenticeship, duplication, and homework) aspects. Second, one becomes a top–level operator only after completing virtually all preceding operator activity phases (e.g., one cannot become a generating unit operator without training as a patrol lineman and boiler or turbine operator, or without having worked successfully in that capacity for a certain time).

The shortcomings of the more popular traditional system include the following:

anyone can apply for an operator job, regardless of whether on not he is psychologically fit for it. This results in a drawn–out natural operator selection process in the course of their unsupervised preparation and training;

while training operators at all levels, the trainees do not have a chance to control a generating unit without the risk of causing undesirable deviations from its set mode, or without a danger of taking a certain action which, in turn, may result in an emergency; and

emergency remediation training [10] does not yield the desired effect since it uses only abstract data which differ from those displayed on a real BShchU SOI, while the trainee does not take any specific actions in the motor field – he reports to the commission orally on what actions he will take and in what sequence.

Training employing a BShchU during a generating unit shutdown is another version of the conventional system. This system is virtually identical to the traditional one except for the fact that the trainee interacts with the BShchU motor field without fearing the consequence of his actions. Yet, as with the earlier case, the TIM is constructed based on a hypothetical situation orally introduced into the exercise by the training instructor (the value of parameters, stop and control valve positions, etc.). The results of the operator interaction with the motor field cannot change the generating unit operation, i.e., the commission judges the propriety or impropriety of his actions on the basis of the trainee's oral answers. There is no real feedback with the object, thus lowering training efficiency.

The first attempt to modify the training principle of generating unit operators was the creation and implementation of a Skill Training Center (UTTs) at the Ukrainian Power Ministry [12] intended for improving and maintaining the skill of 300 MW generating unit operators at Ukrainian TES. The UTTs structure, its instruction and training phase sequence, and the tasks faced at each phase are shown in Fig. 4. The system realized at the UTTs was aimed at eliminating the shortcomings inherent in the traditional system.

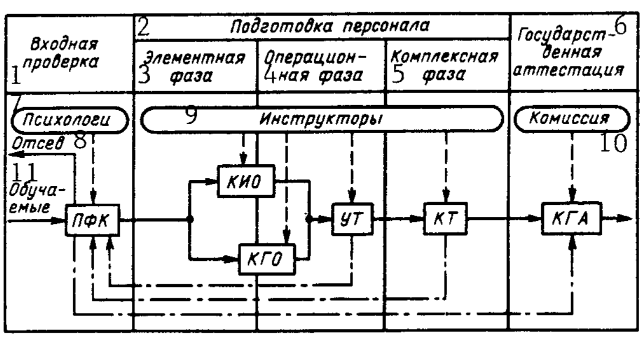


Fig. 4. Operator training system used at the Ukrainian Minenergo UTTs: ПФК – psychophysiological complex; КГА – state certification class; 1 – entrance test; 2 – staff training; 3 – building block phase; 4 – operational phase; 5 – integrated phase; 6 – state certification; 7 – psychologists; 8 – dropouts; 9 – instructors; 10 – commission; 11 – trainees.

During the first phase of his stay at the UTTs, the trainee takes psychophysiological tests (entrance testing) in order to determine his fitness for operator activity. Based on the experience accumulated in the course of UTTs operation, the psychophysiological control lab issues one of the four possible rulings on the basis of the operator applicant's test results: fit, conditionally fit, conditionally unfit, and unfit. In the latter case, the applicant

is dissuaded from engaging in operator activity (in contrast to aviation, space flights, and the Navy, there is still no legislation prohibiting him from being employed as an operator in the power engineering industry). In the former three cases, operator training is expedient. In the first and third case, training programs are adjusted, allowing for the psychophysiological evaluation results. During the second phase, the trainee acquires theoretical knowledge in individual (KIO) and group programmed instruction (KGO) classes in an amount necessary and sufficient for unsupervised work in the control loop. Furthermore, using the so-called algorithmic materials (PERT trees, observation charts, and action plans), the trainee acquires initial operational thinking skills in KIO and KGO. A section (specialized) simulator class (KUT) where the trainee acquires the skill of controlling individual process bays (combustion process, maintaining superheated steam temperature, the generating unit loading process, etc.) – is the third phase of training at UTTs. And finally, during the fourth phase, the trainee acquires the skill of controlling a generating unit using an integrated simulator (KT). The knowledge and skills acquired by the trainee during the preceding phases enable him to use KT with maximum efficiency. Training at the UTTs terminates with taking the exit exam in front of the certification commission.

The UTTs at the Tripolsk GRES has the following shortcomings:

a large percentage of non-standard technological equipment among simulators and training systems;

the limited capabilities of computers used as part of KT make it impossible to realize dynamic models of generating unit components at the quality level required for training; and

the commercial equipment employed in KIO turned out to be inefficient and thus had to be disassembled.

The principal UTTs shortcoming is the fact that the training and instruction process concept had not been refined sufficiently prior to its opening which, in turn, made it impossible to formulate the principal requirements for specific simulators and training systems, for instruction and training scenarios, and for training and teaching aids. Consequently, while mastering the UTTs operation, its staff worked on bringing up to date the simulators, training systems, and their software. Teaching aids and manuals had to be virtually rewritten, and the concept of the instruction/training process in teaching and training systems with a wide use of hardware was developed.

Compared to the conventional system, the principal distinction of the teaching and training system realized at UTTs is the fact that an integrated instruction and training system equipped with special hardware (simulators and tutorial devices) was used for the first time in domestic use, making it possible to check the principal notions embedded in the project as well as to outline its future course.

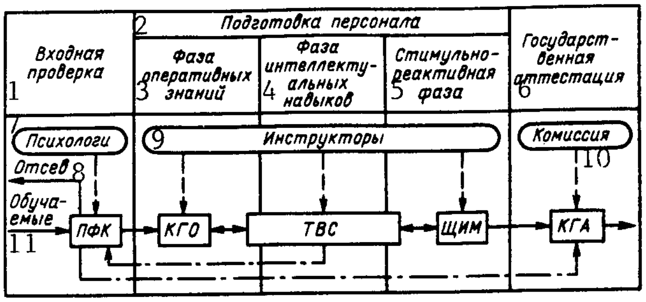


Fig. 5. Conceptual system of operator training [1, 2]: 1 – entrance test; 2 – staff training; 3 – operational knowledge phase; 4 – intellectual skill phase; 5 – stimuli–reaction phase ; 6 – state certification; 7 – psychologists; 8 – dropouts; 9 – instructors; 10 – commission; 11 – trainees.

Conceptual System of Operator Training. In late 1983, the idea of creating the so–called non–conceptual training systems was expressed in [3] and later elaborated on in [1, 2]. The block diagram of the system suggested by the authors of [1, 2] is shown in Fig. 5. By comparing the UTTs structure and the conceptual system project, one can see that KGO are retained in the proposed system while KIO, UT, and KT are absent. They have been replaced with a tutorial computer system (TVS) which, in the opinion of its authors, must insure the transition from knowledge to intellectual skills, while the functions of simulators intended for developing skills and habits are now entrusted to the simulator panels (ShchIM). The latter must help to correlate intellectual skills with actual instruments (SOI) and controls (the motor field).

The methodology of constructing systems is developed in [1, 2] and a detailed description of one such system's realization in TETs–1 of the Krasnoyarskenergo is given. The system described there can be regarded as an integrated system. To settle the argument about which of the aforementioned systems is better, we would probably have to accumulate enough operating experience of the conceptual system as a whole, primarily experience with using TVS.

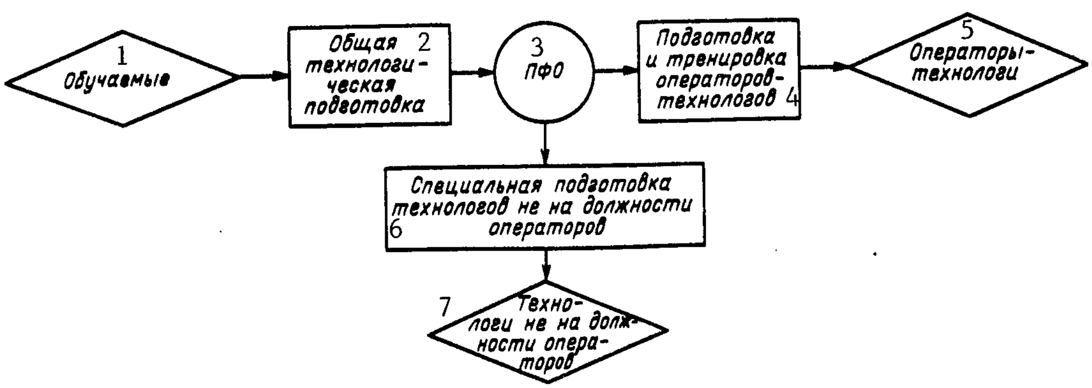


Fig. 6. Structure of a two–phase integrated process engineer/operator training system: 1 – trainees; 2 – general process training; 3 – PFO; 4 – process engineer/operator preparation and training; 5 – process engineers/operators; 6 – specialized process engineer training for other than operator jobs; 7 – process engineers in other than operator positions.

Integrated Two–Phase Tutorial and Training System. A critical review of the tutorial and training process realized at the UTTs at the Tripolsk GRES and an analysis of the results of work with the staff trained by this system made it possible to speculate about the expediency of tentatively dividing the entire operator training process into two phases. During the first phase, highly–skilled process engineers are trained, while during the second – process operators are trained from among the process engineers who have successfully passed the psychophysiological selection process. The trainees who flunked the PFO should be advised to continue training but only for operator positions. The structure of the two–phase tutorial and training system is shown in Fig. 6. The content and structure of both of the system's components – the tutorial system and training system – is described below.

With respect to the foregoing, we can conclude that the training system is intended for preparing highly–skill process engineers. The system must consist of the following three principal components.

1. The tutorial system *per se* is intended to insure the conditions for comprehending general theoretical materials related to understanding the course of the process for the trainee; special aids aimed at studying directives (PTE, PTB, PBYa, etc.).

2. A system to control how the trainees comprehend the aforementioned theoretical, instructional, and specialized materials.

3. A system assisting the trainees to comprehend the interactions and interrelations between the TO status and TP course, as well as the role played by the process operator in the ASU TP.

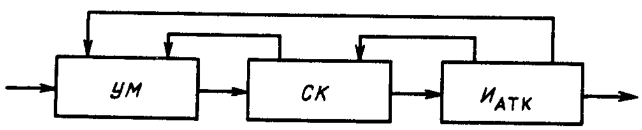


Fig. 7. Tutorial system structure: УМ – teaching aids; СК – control system; $И_{ATK}$ – automatic process system simulator: studying УМ about TO and TP; studying УМ about TO operations; studying directives and standards, studying the interaction of TO and TP.

There are numerous ways of realizing the above systems, ranging from conventional tutorial systems (lectures, seminars, and homework with teaching aids) to automated teaching systems [13]; from traditional testing systems (exams, quizzes, and tests) to automated testing systems; from a theoretical analysis of the interactions and interrelations between TO and TP to studying them with the help of specialized automated benches and industrial system simulators. The system developers and the client selected a certain version based on specific conditions and capabilities (one possible realization version is shown in Fig. 7).

All three of the above systems must correspond to their functional purpose – to insure the conditions for training highly–skilled process engineers.

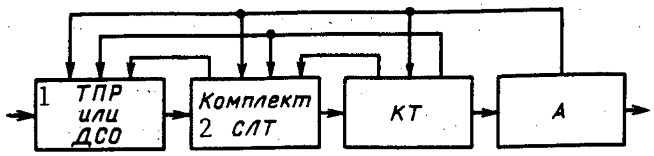


Fig. 8. Simulator training system structure: ТПР – decision-making simulator; ДСО – display training system; САТ – specialized local simulators; КТ – integrated simulator; А – certification; tasks: to develop operational thinking skills, develop control and regulating skills; and develop skills for working in the ASU TP loop; 1 – ТПР or ДСО; 2 – set of САТs.

The simulator training system (Fig. 8) is intended to do the following:

develop operational thinking skills (intellectual skills);

develop skills of controlling individual processes or a group of interrelated processes (elementary skills); and

develop skills for controlling the generating unit under any operating condition.

As with the tutorial system, there are numerous ways of realizing the simulator training system components. Thus, intellectual skills can be developed with the help of the so-called decision-making simulators [2, 4], VDT training systems, and finally – TVS [3]. Likewise, process control skills can be developed with the help of specialized local (bay) simulators [14] or their substitutions executed with the help of commercial SVT [expansion unknown] [15]. And finally, the know-how to control the generating unit as a whole can be developed in an integrated simulator [4] or ShchIM [2, 3]. A method of teaching all of the above skills and habits solely using KT also cannot be ruled out. The decision to select the set of simulators, their type, and structure, and to integrate it into a single training system (one possible integration version is shown in Fig. 8) must be made by simulator system developers and their customers in each specific case. A difference among various designs, structures, and peculiarities of the hardware used in these systems should not become the subject of an argument as to which system is better. The main requirement imposed on simulators as skill and know-how development tools is that they meet psychological and pedagogical demands.

The so-called instruction and training centers (UTTs) and instruction and training stations (UTP) are usually developed on the basis of tutorial and simulation systems.

The UTTs and UTP must be integrated into a single system for teaching and training operators and maintaining their skill level. The UTTs and UTP functions must complement each other.

Instruction and training are primarily intended for basic process operator training and certification. Operator retraining, i.e., training operators with a higher skill level selected from among those with practical work experience as operators, should also be performed

at UTTs. The UTTs structure should meet the aforementioned demands. The two-phase integrated system may be recommended as one version of realizing the UTTs structure.

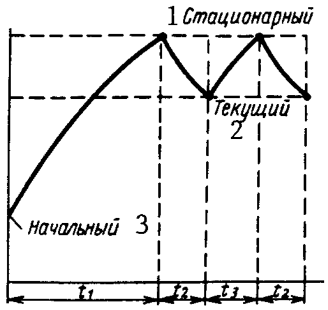


Fig. 9. The process of acquiring, losing, and regaining skills (according to B. A. Smirnov); t_1 – initial learning curve time; t_2 – skill loss time; t_3 – skill recovery time; 1 – stationary; 2 – current; 3 – initial.

The UTP's principal purpose is to insure conditions for maintaining the professional competence level of working operators. The skill acquisition, loss, and recovery diagram (a similar diagram probably characterizes the know–how acquisition, loss, and recovery process) borrowed from [16] is shown in Fig. 9. One can see from the diagram that while working with a real controlled object, skills (know–how) are lost due to the fact that some of them are not used for a certain period of time or used seldom. This is typical of all types of skills – from information perception skills and the ability to "see" the whole picture to the skills of interacting with TOU [expansion unknown] by affecting it through monitoring and control systems.

Thus, the need to develop a system for maintaining the professional readiness of operators is evident; such a system should include the following:

a full–scale integrated simulator;

a special purpose tutorial and testing system;

a system for monitoring current psychophysiological operator characteristics, including pre–shift evaluations;

an automated reference (maybe even game) system, enabling the operator to obtain exhaustive information, analyze it, and understand the causes of existing errors; and

an automated, continuously updated on–line reference system enabling him to obtain data on the status of a specific TOU, ASU TP, or their components at any time.

The structure of a specific UTP is dictated by the customer and developer capabilities, and may vary from a single integrated simulator to all of the above systems in the UTP.

The components of both training and simulation systems and the system as a whole must primarily correspond to their functional purpose, i.e., insure high performance reliability of operators.

In the authors' opinion, this is the principal conclusion to be drawn from this article.

BIBLIOGRAPHY

1. Dyakov, A. F., Venda, V. F., Magazanik, Ya. M. "Metodicheskiye rekomendatsii po sozdaniyu sistem obucheniya i trenazha dlya podgotovki, perepodgotovki i podderzhaniya urovnya natrenirovannosti operatorov energoblokov TES i AES" [Methodological recommendations for developing instruction and training systems for training, retraining, and maintaining skill levels of generating unit operators at TES and AES]. Krasnoyarsk, 1985.

2. Dyakov, A. F., Garbar, S. D. Problem-oriented operator training hardware. ELEKTRIChESKIYe STANTsII, 1988, No. 7.

3. Chachko, A. G. Selecting nuclear power plant operator training principles. ELEKTRIChESKIYe STANTsII, 1983, No. 12.

4. Tsiptsyura, R. D. Integrated instruction and training system for TES and AES generating unit operators. ELEKTRIChESKIYe STANTsII, 1985, No. 8.

5. Babikov, V. M., Panasenko, I. M. Analysis of AES function disruptions and the task of improving operator training. ATOMNAYa ENERGETIKA ZA RUBEZhOM, 1981, No. 9.

6. Kotik, M. A. "Kurs inzhenernoy psikhologii" [Engineering psychology]. Tallinn: Valgus, 1978.

7. "Spravochnik po inzhenernoy psikhologii" [Engineering psychology handbook]. Edited by B. F. Lomov. Moscow: Mashinostroyeniye, 1985.

8. Fedotov, V. K., Melnik, L. I. Experimental investigation into the generating unit operator's work load. ELEKTRIChESKIYe STANTsII, 1976, No. 7.

9. Iovenko, O. V., Chachko, A. G. Simulation of operator decisions in evaluating a work situation. PRIBORY I SISTEMY UPRAVLENIYa, 1974, No. 6.

10. "Rukovodyashchiye ukazaniya po organizatsii raboty s personalom na energopredpriyatiyakh i v organizatsiyakh" [Directive for working with a power generating enterprise and organization staff]. Moscow: Enegoatomizdat, 1983.

11. "Yedinaya metodika podgotovki i provedeniya protivoavariynykh trenirovochnykh ucheniy personala elektricheskikh stantsiy i setey" [Unified methodology for preparing and conducting an emergency remediation training exercise with electric power plant and network personnel]. Moscow: STsNTI ORGRES, 1972.

12. Training/instruction center at the Ukrainian Energy Ministry for training 300 MW generating unit operators. ELEKTRIChESKIYe STANTsII, 1981, No. 10.

13. Sklyarov, V. F., Samoylov, Ye. V., Kolesnikov, S. V., Romanenko, Ye. V. Computer–based professional classification structure of the industry. ELEKTRONNOYe MODELIROVANIYe, 1987, No. 2.

14. Tsiptsyura, R. D., Dosenko, A. D., Svezhentsov, V. A. Bay simulators – an integral component of the simulator complex at a training and instruction center. ENERGETIKA I ELEKTRIFIKATsSIYa, 1985, No. 3.

15. Pisarenko, A. P., Samoylov, V. D. A formalized description of models of TES and AES simulator equipment. ELEKTRONNOYe MODELIROVANIYe, 1984, No. 6.

16. Smirnov, B. A., Dushkov, B. A., Kosmolinskiy, F. P. "Inzhenernaya psikhologiya. Ekonomicheskiye problemy" [Engineering psychology: Economic aspects]. Moscow: Ekonomika, 1983.

UDC 621.313.322.81:621.313.39.3.016.31.003.1

Controlling the Parallel Operation of Synchronous and Asynchronized Turbogenerators

18610459B Moscow ELEKTRICHESKIYE STANTSII in Russian No 3, Mar 89 (signed to press 22 Feb 89) pp 24-28

[Article by Engineers V. P. Oleksin and A. I. Matviychuk and Candidate of Technical Sciences A. S. Minyaylo, Lvovenergo Economic Planning Division and the Lvov Planning Institute]

[Text] The reactive power duty of a turbine-driven generator generally affects active power losses, the dynamic stability limit condition, and the generator's thermal condition [1, 2]. These indicators, in turn, are economic efficiency and reliability components of the electric power plant as a whole.

Today, voltage regulation on electric power plant buses with conventional synchronous turbogenerators (STG) is realized by the optimal group control method [3]. In so doing, for similar generators with identical relationships between their principal parameters and their reactive power, given an equal active power, the equality of their reactive power is the optimal duty condition.

At today's stage of the USSR Unified Power Grid (YeES) development, it is expected that asynchronized turbogenerators (ASTG) will be used at power plants along with STG [1, 4]. Parallel operation of STG and ASTG will be characterized by a number of features determined by peculiarities of ASTG characteristics and properties. It follows from the theory and practice of asynchronized generators [1, 5, 6] that STG and ASTG differ significantly in losses, dynamic stability limit conditions, and thermal rotor conditions as a function of reactive power, given identical active power. Consequently, an equalization of STG and ASTG reactive power at the same active power does not insure optimal voltage control on power plant buses.

The problem of developing a technique for optimizing and controlling the reactive power duty of electric power plants containing STG and ASTG is solved in this article. In so doing, the conditions and basic principles of insuring and realizing optimization and con-

trol of parallel STG and ASTG operation, engineering and economic aspects of this task, and its peculiarities are examined and analyzed.

In order to optimize the parallel operation of STG and ASTG by manipulating their reactive power, one must take into account not one, but several important criteria. These include the following:

insuring a minimum of active power losses in the STG–ASTG system;

insuring maximum reliability of the STG–ASTG system with respect to dynamic stability conditions; and

insuring maximum reliability of the STG–ASTG system with respect to the thermal condition of generators.

The stated optimization problem calls for using a multicriterion approach [7]. One way of meeting these requirements is to convert each criterion into hourly outlays

$$\text{З} = C\text{з},$$

where C is the criterion's value of the optimized entity (OBO) and з is the cost (price) of the criterion's OBO unit.

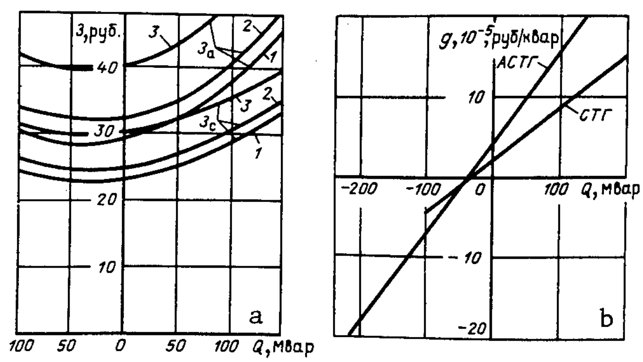


Fig. 1. Plots of total outlays (a) and averaged relative costs increments (b) as a function of reactive power for TGV–200 and ASTG–200 units, in rubles per quarter: *1* – 140 MW; *2* – 160 MW; *3* – 200 MW.

Thus, for each combination of a turbogenerator and a generating unit transformer, we form a weighted average criterion – to insure a minimum of outlays related to reactive power transmission

$$\text{З}_{\text{б}} = \text{З}_{\text{п}} + \text{З}_{\text{д}} + \text{З}_{\text{тс}} + \text{З}_{\text{т}}, \qquad (1)$$

where $\text{З}_{\text{п}}$, $\text{З}_{\text{д}}$, $\text{З}_{\text{тс}}$, and $\text{З}_{\text{т}}$ are outlays for recovering active power losses in the generator, losses from a drop in dynamic stability and fuel overruns during the unit's dynamic operation, losses from a drop in generator reliability according to its thermal condition, and losses to compensate for active power losses in the unit's transformer, respectively. Using formula (1), functional relationships between the outlay $\text{З}_{\text{б}}$ and reactive power at certain

active power values was calculated for the TGV–200 and ASTG–200–2UZ. These relationships are shown in Fig. 1a. In so doing, the $З_{тс}$ loss component was ignored, since there are still no reliable quantitative relationships between the generator reliability and its operating parameters. An examination of the derived relationships attests to the fact that they differ significantly for STG and ASTG. Given a system of n units with STG and ASTG, the total outlay criterion function assumes the form of

$$З = \sum_{i=1}^{n} З_i \to \min,$$

where $i = 1, 2, ..., n$ is the unit's ordinal number. It is easy to prove [8] that the reactive power optimality condition of the system under consideration is

$$\frac{\partial З_1}{\partial Q_1} = \frac{\partial З_2}{\partial Q_2} = \ldots = \frac{\partial З_n}{\partial Q_n}$$

or

$$q_1 = q_2 = \ldots = q_n \ . \tag{2}$$

In the final analysis, calculation of the marginal costs related to transmitting reactive power is reduced to differentiating the function $З = f(Q)$. The difficulty of resolving this issue is due to the fact that it is not always possible to derive a precise mathematical expression of the outlay function under consideration. Moreover, a discrete OBO function $C = f(Q)$ and, consequently, the expenditure function $З = f(Q)$ for each individual criterion of the turbogenerator and generating unit transformer and, in the final analysis, the total unit expenditure function according to formula (1) as a whole, can be obtained with the help of existing computational techniques or testing. Today, spline methods are widely used for approximating discrete functions. They enable us to approximate the discrete function of generating unit expenditures $З_б = f(Q)$, while subsequent interpolation and differentiation – to determine relative incremental expenditures at each m^{th} point

$$q_{\nu m} = a_{1\nu m} + 2a_{2\nu m} Q_{\nu 1 m} + \ldots + t a_{t\nu 1 n} Q_{\nu m}^{t-1} ,$$

where $m = 1, 2, ..., r$ is the number of discrete reactive power values assumed; and t is the power of the assumed polynomial.

By making calculations for the entire feasible area of generator operations, we obtain the discrete function of relative incremental expenditures of the unit

$$q_{pr} = f(Q_{pr}), \tag{3}$$

where q_{pr} is a two–dimensional array of relative expenditures with $p \times r$ dimensions; and Q_{pr} is a two–dimensional array of assigned reactive power values with $p \times r$ dimensions; and p is the number of active power levels assumed.

In practice, the optimum reactive power distribution among turbogenerators is difficult to realize by using two–dimensional relative increment arrays. Consequently, let us simplify discrete function (3) by reducing it to a simpler dependence. To this end, let us use the least squares method enabling us to average discrete dependence (3) of the i^{th} unit to a continuous linearity of the

$$q_i = a_i Q + b_i \tag{4}$$

type, where a_i and b_i are the corresponding coefficients.

Based on the above analysis, the algorithm and program for computing real values and averaged rectilinear characteristics of relative incremental outlays of generating units with STG and ASTG were compiled in the FORTRAN–IV language. This program's realization in the YeS 1045 digital computer (ETsVM) made it possible to determine relative incremental outlays for the TGV–200 and ASTG–200 turbogenerators with unit transformers. Rectilinear graphic plots of $q = f(Q)$ for STG and ASTG drawn on the basis of the calculation results are shown in Fig. lb. A simultaneous analysis of the rectilinear characteristics of relative increments and their real values attest to the fact that the maximum deviation from a linear relationship is 3.33% for the TGV–200 and 8.68% for the ASTG–200 relative to their values at rated duty. It occurs in the maximum reactive power yield area. At the center of the control range, as well as in the reactive power consumption operation, the relative increment deviation from a linear relationship does not exceed 2% for both generators under study.

Let us assume that relative increments of the generators are linear characteristics which do not depend on the active power loading of the units. The order of optimal total reactive load distribution between STG and ASTG in such a case is shown in Fig. 2a. In essence, it amounts to forming the functions of relative increments of a system made up of STG and ASTG generating units, namely:

$$q_{H} = f(Q_{H}). \tag{5}$$

The optimal distribution is based on the condition of marginal cost equality, consequently, the scales of q generating units and their system (loads) have the same graduation. The scale of the functional dependence of the system of units Q_H is formed by adding up STG and ASTG reactive power

$$Q_H = Q_c + Q_a \tag{6}$$

if condition (2) is met.

Thus, in order to derive relation (5), the generator's relative reactive power is added up. Consequently, given a said value of reactive load Q_{H3}, it is easy to determine the optimal reactive power of STG and ASTG in a reverse sequence.

Let us consider the basic premises of optimal distribution of the total reactive load between STG and ASTG units based on using real functional relations of relative increments. According to expressions (2), (5), and (6), in order to realize the optimal reactive power distribution between STG and ASTG, it is necessary that like terms of relative increment arrays of generators under study be identical. This is attained by interpolating discrete functions (3) successively for each fixed reactive power value, from the starting value of $q_н$ to the final $q_к$ value of relative increments with a step of Δq. The newly derived discrete function of the turbogenerator's relative increments assumes the form of

$$q_r = f(Q_{pr}), \tag{7}$$

where r is the number of discrete relative increment values assumed.

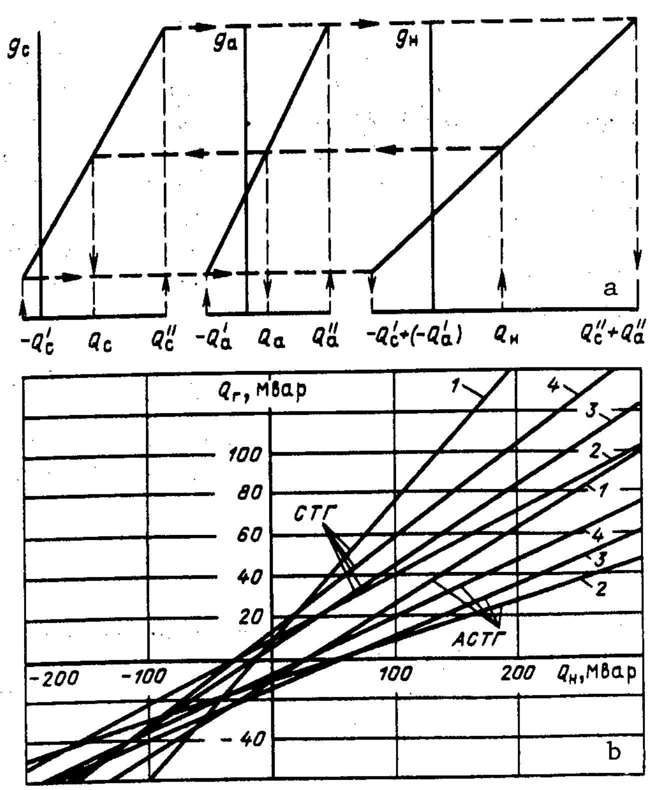


Fig. 2. Plots of optimal total reactive load distribution between STG and ASTG: a – general distribution patterns; b – plots of optimal reactive STG and ASTG power given various generating unit makeups: *1* – 1 × STG and 1 × ASTG; *2* – 2 × STG and 2 × ASTG; *3* – 2 × STG and 1 × ASTG; *4* – 1 × STG and 2 × ASTG.

Based on expressions (7), as well as (2), (5), and (6), we obtain the discrete dependence of the function of optimal relative STG and ASTG power on the generators' total reactive load:

$$Q^{c}_{pr} = F(Q^{н}_{pr}); \tag{8}$$

$$Q^{a}_{pr} = F(Q^{н}_{pr}). \tag{9}$$

The resulting arrays of optimal reactive generating unit power are used directly in making the optimal distribution. To this end, the generator power is determined by a given reactive load from relations (8) and (9). The corresponding algorithms developed for this purpose make it possible to calculate the generators' optimal reactive power using a computer.

The total reactive power of a group of n generating units is

$$Q_1 + Q_2 + \dots + Q_n = Q_H. \tag{10}$$

Allowing for condition (2), expressions (4) and (10) taken together enable us to derive a linear dependence of the optimal reactive power of the i^{th} unit on the total reactive load

$$Q_i = C_i\, Q_H + D_i\,, \tag{11}$$

where

$$C_i = \frac{1}{a \sum_{i=1}^{n} \frac{1}{a_i}}\,; \quad D_i = \frac{\sum_{i=1}^{n} \frac{b_i}{a_i} - b_i \sum_{i=1}^{n} \frac{1}{a_i}}{a_i \sum_{i=1}^{n} \frac{1}{a_i}}.$$

From an analysis of the resulting dependence (11), it follows that the generator's optimal reactive power is a function of the following operational factors:

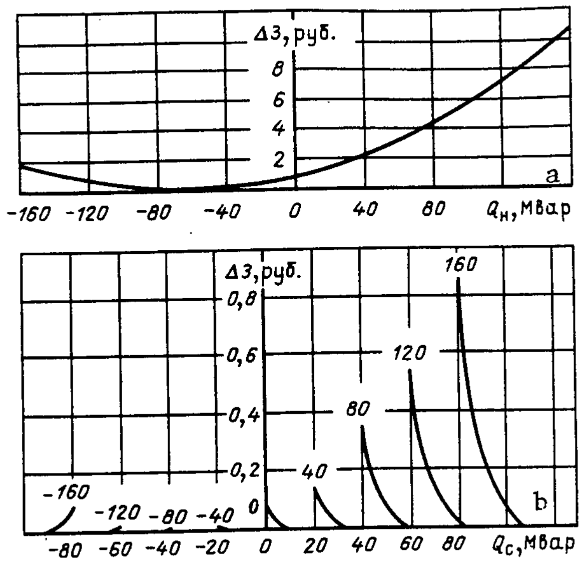
the power plant's total reactive power; and
specific makeup, rather than percentage ratio, of generating units.

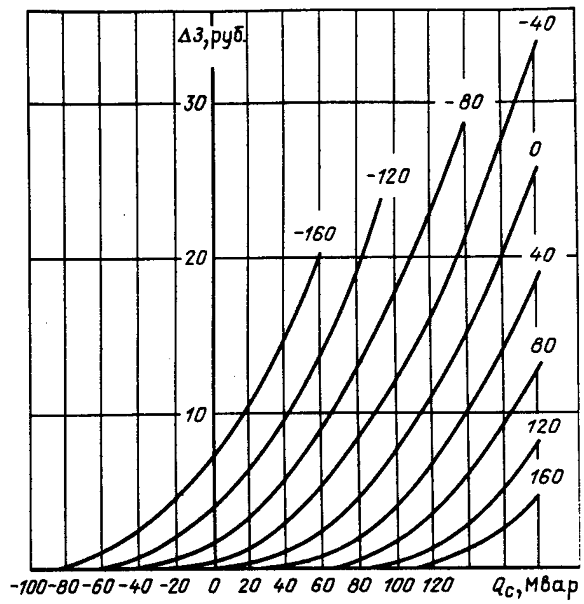
For illustration, (11)–type relationships were computed for a number of generating unit groups consisting of the TGV–200 and ASTG–200; their plots are shown in Fig. 2b. It follows from the relations thus derived that in order to insure a minimum of outlays in the ASTG system under study compared to the STG system, we are forced to shift toward an area of lower reactive power and its smaller consumption from the grid.

The use of real characteristics of relative increments can be made effective for maintaining optimal power plant operations with STG and ASTG by using a computer.

The use of averaged linear characteristics of relative increments may significantly streamline the process of optimal total reactive load distribution between STG and ASTG. To this end, it is expedient to develop optimal group voltage control on the busbars of electric power plants containing STG and ASTG.

The capabilities of various methods of controlling the turbogenerators' reactive power [3, 6] calls for evaluating the proposed optimal reactive power distribution between STG and ASTG.





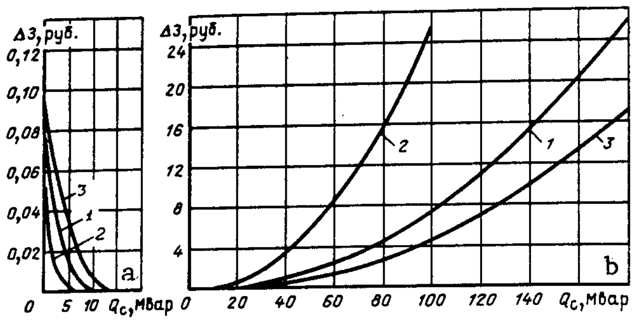


Fig. 3. Plots of hourly expenditure variations of 1 × TVG–200 – 1 × ASTG–200 units under various loads: a – optimal operation; b – given an operation deviation toward reactive power equality of both units (top left).

Fig. 4. Plots of hourly expenditure variations as a function of power characterizing the operation deviation from the optimum for the 1 × TVG–200 – 1 × ASTG–200 combination of units under reactive load (top right).

Fig. 5. Plots of hourly expenditure variations as a function of power characterizing the operation deviation from the optimum given a null total reactive load for various generating unit makeups: a – deviation toward reactive power equality of STG and ASTG; b – deviation toward increasing reactive STG load; *1* – 1 × STG – 1 × ASTG; *2* – 2 × STG – 1 × ASTG; *3* – 1 × STG – 2 × ASTG (bottom left).

To this end, we calculated the total expenditure variation of a system of STG–ASTG units on the basis of the computation program compiled for this purpose and dependence (11), given a duty deviation from the optimum for various combinations of generating units and total reactive loads. The difference between the computed relationships for various values of active generator power differ insignificantly but do not exceed 0.34 ruble in operations characterized by the greatest expenditure deviation and 0.006 ruble in oper-

ations approaching the optimum. Moreover, the greatest difference pertains to lower active power of the unit. For illustration, the plots of expenditure variations with duty deviations from the optimal operation are shown in Figs. 3, 4, and 5. One can see from analyzing these plots that the use of STG and ASTG is the most efficient for covering small reactive loads, especially when taking up power from the grid. A condition's deviation from the optimum increases the total expenditures related to transmitting reactive power. Moreover, this especially significant when the STG reactive power loading is increased (switching the ASTG to a deep consumption mode).

To estimate the efficacy of the proposed optimal reactive load distribution method, we examined the operating conditions of the TGV–200 and ASTG–200 in a state regional electric power plant (GRES) where the first production prototype asynchronized turbine-driven generator was implemented during 1982–1985 on the basis of analyzing representative daily electric load schedules. In so doing, we used the computational program and relation (11) to calculate the power plant expenditure variation during the day for each representative daily schedule under existing and proposed methods of reactive power distribution between STG and ASTG. The resulting relationships are adjusted for one year. An analysis of the results thus obtained attests to the fact that the reactive power distribution between STG and ASTG is 14.02 thousand rubles a year per one ASTG compared to other known techniques.

Summary

1. The reactive power optimization of the STG and ASTG collaboration requires multiple criteria and calls for using a weighted average criterion – insuring a minimum total outlay of the generator/generating unit transformer combination. The reactive power loading of a turbogenerator is optimal if their relative increments, determined by approximation with subsequent interpolation and differentiation of the total criterion function of expenditures with the help of the spline method, are equal.

2. To control optimal STG and ASTG operations, it is expedient to select the dependence of the generators' total reactive power on the power plant's total reactive load. Moreover, the use of real relationships is most efficient if computers are used, while the averaging of linear characteristics – in developing a special optimal method of group voltage control on the busbars of electric power plants containing STG and ASTG.

3. The proposed technique for optimizing the parallel operation of STG and ASTG is economically preferable to other possible control methods. The economic gain realized from using it depends on operating conditions and the numerical composition of generating units with STG and ASTG.

BIBLIOGRAPHY

1. Shakaryan, Yu. G. "Asinkhronizirovannyye sinkhronnyye mashiny" [Asynchronized synchronous generators]. Moscow: Energoatomizdat, 1984.

2. Kostenko, M. P., Piotrovskiy, L. M. "Elektricheskiye mashiny" [Electric generators and motors]. Part 2. Moscow–Leningrad: Gosenergoizdat, 1958.

3. Rosman, L. V. "Sostavleniye i raschet sistem gruppovogo upravleniya vozbuzhdeniyem sinkhronnykh generatorov" [Design and analysis of group synchronous generator excitation control systems]. Moscow: Energiya, 1986.

4. Krivushkin, I. F., Chevychelov, V. A. On feasible applications of asynchronized turbogenerators. ELEKTRIChESKIYe STANTsII, 1983, No. 10.

5. Chevychelov, V. A. Performance characteristics of electric power plants with asynchronized turbogenerators. ELEKTRIChESKIYe STANTsII, 1986, No. 8.

6. Prokofyev, Yu. A., Tsviliy, S. L., Bogoslovskiy, Yu. M. "Sovmestnaya rabota asinkhronizirovannykh i sinkhronnykh turbogeneratorov. Sovremennyye problemy energetiki. Tezisy doklada IV Respublikanskoy nauchno–tekhnicheskoy konferentsii" [Parallel operation of asynchronized and synchronous turbogenerators. Today's problems in the power industry. Proceedings of the Fourth Republican Scientific and Engineering Conference]. Part 2. Kiev, 1985.

7. Venikov, V. A., Zhuravlev, V. G., Fillipova, T. A. Optimizatsiya rezhimov elektrostantsiy i energosistem" [Optimization of electric power plant and power grid operation]. Moscow: Energoatomizdat, 1981.

8. Syromyatnikov, I. A. "Rezhimy raboty asinkhronnykh i sinchronnykh dvigateley" [Operations of synchronous and induction motors]. Edited by L. G. Mamikoyants. Fourth edition, revised and expanded. Moscow: Energoatomizdat, 1984.

UDC 681.3.06

On Building a Heterogeneous Local Computer Network for Developing Flexible Manufacturing Systems

18610428B Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 6, Nov-Dec 88 (manuscript received 28 Sep 87) pp 92-95

[Article by N. P. Starodub, A. I. Slobodyanyuk, and S. D. Pogorelyy]

[Text] Today's flexible manufacturing systems (GPS) are characterized by their diverse automated equipment. This equipment, in turn, contains a broad range of both imbedded and off-line computer hardware. The functioning of GPS presents the task of integrating various computer hardware into heterogeneous local networks. The structure and software of a heterogeneous computer (EVM) network containing SOU-2 minicomputers are described below.

Network Hardware. A local network's hardware includes the following: a YeS computer, two SOU-2 minicomputers, an SM-1420 minicomputer, a UVS-01 microcomputer, an A71118 computer interface (USVM), an ADS-4 long-distance communication interface (BS), a complex of network controllers (SK) for SOU-2 minicomputers, and a UVS-01 microcomputer.

The heterogeneous local network is built according to the packet switching principle with a fixed route selection strategy and have a "star" type topology. This topology is characterized in that two subscribers communicate through a single central (transport) node executed as a SOU-2 computer.

A block diagram of the heterogeneous local network hardware is shown in Fig. 1. The "star" type topology makes it possible to realize a three-layer hierarchy. The YeS computer which has the greatest capacity and external memory is used as layer III. This makes it possible to use the YeS computer to solve problems requiring considerable execution time outlays and for setting up databases. The layer III computer has vertical communications with layer II computers which have vertical communications with upper and lower layer computers, as well as horizontal communications within the layers.

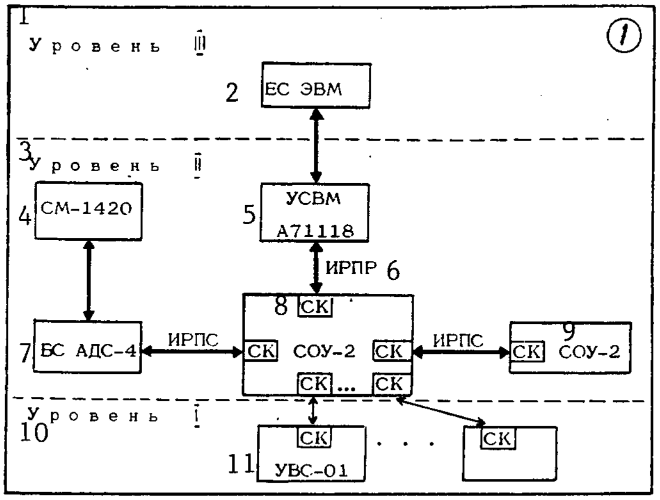


Fig. 1. Heterogeneous local network hardware topology: 1 – layer I; 2 – YeS computer; 3 – layer II; 4 – SM–1420 computer; 5 – USVM; 6 – IRPR; 7 – BS ADS–4; 8 – SK; 9 – SOU–2; 10 – layer III; 11 – UVS–01.

The hardware interfacing two SOU–2 minicomputers represents network controller modules installed in these minicomputers and performing an intercomputer communication over an IRPS interface. The data interchange rate between these computers is 9,600 bit/s, given a maximum distance of up to 500 m.

The hardware interfacing SOU–2 minicomputers with the UVS–01 microcomputer contains network controller modules installed in these computers. The intercomputer communication over an IRPS interface is realized at a 9,600 bit/s rate.

The hardware interfacing the SOU–2 minicomputer with the SM computer is a network controller on the SOU–2 minicomputer side and an ADS–4 interface unit (a commercial SM computer component) on the SM computer side. The intercomputer communication over an IRPS interface is realized at a 9,600 bit/s rate.

The hardware interfacing the SOU–2 minicomputer and the YeS computer is an off–the-shelf A71118 computer communication device on the YeS computer side and network controllers on the SOU–2 side. The data interchange rate over a parallel interface (IRPR) in this device is at 250 kbit/s.

Several microcomputers distributed among various entities of flexible manufacturing modules performing the functions of data collection and preprocessing and direct digital control can be used at the lower layer. Thus, the number of microcomputers at the lower layer may be increased and the functional capabilities of the heterogeneous local network expanded.

Network Software. The network software package (Fig. 2) is a combination of software distributed over the local network's nodes (computers) and placed in a proper environment. The local network's environment is formed of computer operating systems incorporated in the network: OS RV 3.0 for the SM computer, OS RVSP for "Neyron–Kh", SR/M2.0 for "Neyron–I", and OS YeS 6.1 for the YeS computer.

The network software package has a modular structure and consists of the following components:

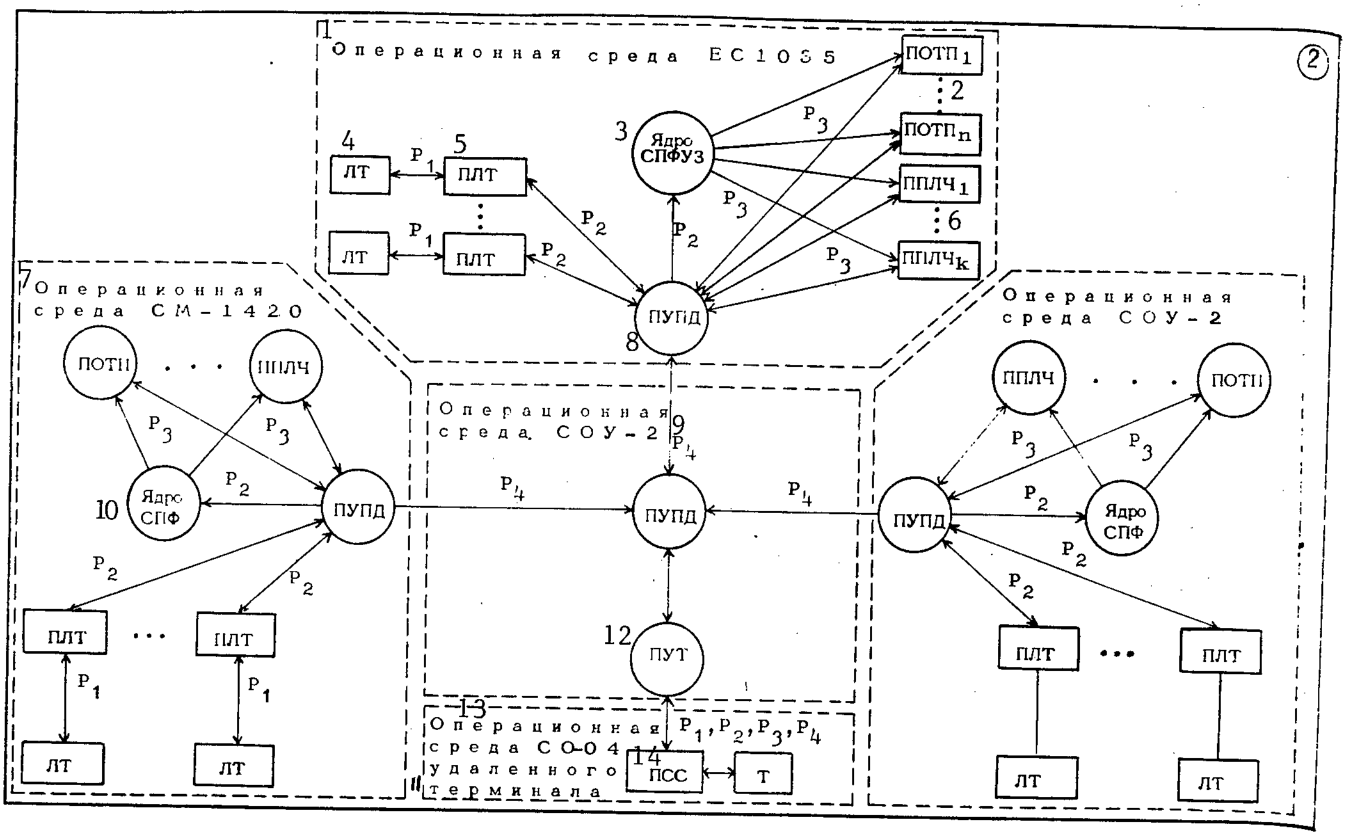


Fig. 2. Block diagram of the network software package for a heterogeneous computer network: 1 – YeS 1035 environment; 2 – POTP; 3 – SPFUZ kernel; 4 – LT; 5 – PLT; 6 – PPLCh; 7 – SM–1420 environment; 8 – PUPD; 9 – SOU–2 environment; 10 – SPF kernel; 11 – remote terminal; 12 – PUT; 13 – SO–04 environment; 14 – PSS.

- data transmission control programs (PUPD);
- file and remote job transmission system SPFUZ; and
- SPFUZ network terminal programs.

Depending on the role played by each computer in the local network, its functional loading with programs from the network software package will be different. Communication computers need only data transmission control programs; terminal computers – network terminal programs; terminal/communication computers – data transmission control and

network terminal programs; and the mainframes, which have a high output and a large RAM and disk memory, require all three groups of programs.

In addition to the above components, the network software package contains a set of programs for testing communication channels for each adjacent pair of computers.

Data transmission control programs perform data interchange among network nodes in the form of batches (datagrams) consisting of a batch header and an information part. The size of the batch's information part must not exceed the maximum defined by the total ideology of data transmission in the network or the type of physical medium. Data transmission control program users exchange letters whose size may exceed that of the information part. Consequently, to be transmitted a letter is broken up into several fragments so that the size of each fragment, except the last, is equal to the maximum size of the batch's information part; then the data transmission control program sends the letter from the sender toward to the addressee fragment by fragment. When the interchange process has been completed, the sender and the addressee are so informed by the corresponding data transmission control programs.

Batches are transmitted over fixed routes in the network through the nodes (computers) and output communication channels determined beforehand to the next computer along the route. Batch routes are logged in route tables for all data transmission control programs during the network topology tuning phase.

Data transmission over communication channels between adjacent computers is monitored. Upon detecting an improperly transmitted file, it is retransmitted.

Query blocks of a fixed length with the following information are used as input data for data transmission control programs: an identifier of the job (user) making the query; the query code; and query arguments.

The query code determines the actions which must be taken by the data transmission control program: e.g., to make a connection, disconnect, open a port, close a port, send a letter, announce a receiving buffer, and scan the job's network addresses.

Query arguments define the parameters necessary for executing the actions indicated by the query code: e.g., the data transmission control program user number, port number, receiving buffer number and size, and distant user network address.

The data transmission control program's output data are return codes about the results of executing the actions assigned by the query code, letter control blocks, and receiving buffer control blocks.

File and remote job transmission service programs are intended for exchanging the content of files among network computers separated over a large territory and, in turn, are composed of a control module and a group of sender and addressee processing modules

whose number is determined by the RAM size and the operating system's capabilities of creating and timing parallel processes.

The control module enables the system to receive user queries by enabling the user to schedule and control the computational process, manages the processing, and maintains the user query timing mechanism.

User queries represent the following primitives: sender and addressee process formation, data transmission, and timing. Query processing includes analyzing it, determining the executor, making a call, and exchanging control and information messages containing the query and the results of its execution. The timing mechanism being maintained makes it possible to identify the incoming queries, check their execution status, and provide proper information to the user.

Using a completed call to the control module, processing modules receive queries for transmitting data and placing them in disk files, organize interchange between the "sender and addressee" pairs, and insure the transmission of information about the completion of the query to the party initiating the interchange.

On the YeS computer side, the addressee's processing module and the control module are equipped with the means to input, start, and scan the remote job status.

Network terminal programs of the file and remote job transmission service enable users to access the file banks of the entire network and organize the file and remote job transmission itself using network terminal resources. An interface with the user represents a conical form of file operation primitives performing the following functions:

- file and remote job transmission;
- controlling standard OS YeS functions using operating commands; and
- receiving reference data about the network.

Network terminal program algorithms provide for realizing the "trilateral transmission" mode, indicating the presence of a transmission originator, a file sender, and a file recipient in the network.

The originator assumes the function of coordinating the file transmission between the sender and the recipient. Moreover, the transmission participants may generally be located at different network nodes (computers).

The network terminal program contains two independent groups: local terminal programs (PLT) and remote terminal programs (PUT). A local terminal (LT) is understood as a terminal embedded in the configuration of each computer's peripheral, while the remote terminal – as a microcomputer of any architecture with a network communication module (PSS) contained in its software. An SO–04 or SO–05 microprocessor debugging system with the SR/M0 operating system is used as the remote terminal.

With only a small refinement, the network terminal program algorithm makes it possible to expand embedded functions, e.g., view a remote file at a terminal, prepare a file at a terminal with subsequent transmission to any network center, etc.

Four groups of protocols have been developed to insure proper functioning of all network software components:

- a protocol between the user and network terminal program (P_1);
- a protocol between the network terminal program and file and remote job transmission service kernel (P_2);
- a protocol between the sender and the recipient (P_3); and
- a protocol between data transmission programs of all network computers (P_4).

The complex of network software packages executed collaterally by all network computers is the basic resource for developing distributed data processing and storage systems for flexible automated manufacturing and insures the resources for exchanging messages of various lengths between a remote pair of tasks, for file transmission, and for remote interactive job input and processing from any local network computer.

The above hardware and software complex makes it possible to build heterogeneous local computer networks. A section of numerically controlled (ChPU) machine tools was automated with the help of these resources. In this GPS, the task of designing programs for machine tools is performed by YeS and SM–1420 computers, the programs are simulated and debugged, then transmitted over a communication channel to the UVS–01 microcomputer whence they are transmitted to machine tool control racks. Local network implementation increases GPS output and reliability by a factor of 1.5–2.

Today, a second local network subsystem (without the YeS computer) is being implemented at the debugging system section. An SOU–2 computer generates control and diagnostic tests which are transmitted to debugger workstations equipped with UVS–01 microcomputers.

UDC 621.9.06

Language Tools for Planning the Control of Advanced Electronic Automatic Equipment

18610428C Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 6, Nov-Dec 88 (manuscript received 14 Jul 86; revised version received 7 Dec 87) pp 95-100

[Article by V. L. Sosonkin and L. Ye. Shergin]

[Text] Resources for controlling advanced electronic automatic equipment of numerically controlled (ChPU) machine tools, robots, manipulators, retooled lines, transfer machines, shipping and warehousing systems, etc., play an important role in real control systems. The programming simplicity and reliability of electronic automatic equipment control systems, the program debugging convenience, and the diagnostics and retuning of the ongoing control processes are becoming increasingly important.

The aim of this paper is to describe the language tools for designing the process control of complicated electronic automatic equipment for multifunctional purposes. These tools are based on the improved problem-oriented background controlled (FOKON) language, its compiler, and debugger. The term "background" emphasizes the active role of software in controlling electronic automation equipment against the background of other tasks of the control device.

In developing a real time problem-oriented language, the following problem execution sequence was adopted:

- shaping the internal problem-oriented structure of real time electronic automation equipment control programs, including allowing for machine codes of the chosen processor (at first the UChPU 2S42 processor of the "Elektronika-60" microcomputer was used as such);
- determining the structure of the language tools representing the control process of electronic automatic equipment which matches the selected underlying program structure and has a naturally logical structure from the language user viewpoint;

- developing a corresponding set of operators (primitives) insuring the interaction of individual real time control processes of electronic automatic equipment;

- forming a set of logical and computational operators for working with the data structures used;

- developing forms of imbedding electronic automatic equipment control programs into main programs of the control system and developing interaction methods of all programs; and

- developing means of designing, graphically displaying, debugging, and diagnosing electronic automatic equipment control programs.

The FOKON language is aimed at describing and checking out the processes of electronic automation in ChPU systems and programmable I/O chips separately for various product units, allowing for control and data constraints. Within the processes, the unit operation is described from one status/condition to the next status/condition. A condition is a program repeated at a fixed rate for a given process which describes the conditions being analyzed at a given unit status and the actions to be taken when these conditions are reached. Given a conditional or unconditional branch to another process situation, this new condition becomes active starting at the next cycle of a given process. Thus, only one program, called the active process condition, is repeated for any process at each time period.

The control process structure is represented as follows. The structure is based on a sequence of conditions with a set of mutual branches. A set of input programs for receiving queries is outside this base. Furthermore, the base (sequence of conditions) is represented as a set of condition blocks. Before determining the process interaction forms being entered, let us clarify the set of control processes allowing for their priority layers:

- processes whose busy conditions are executed by the interrupts originating them, disabling other interrupts;

- processes whose busy conditions are only designated for execution by the corresponding interrupts, but are executed at process layers of other priorities to which they are compared;

- processes whose busy conditions are executed by a single group at a rate of, e.g., 100 Hz; the process execution sequence is constant in this group; before executing processes in this group, the queue of incoming queries from higher priority processes to the input receiving programs and jobs for interrupt process programs compared to this priority layer is submitted;

- processes which are similar to the proceeding ones but have a lower priority are submitted at a rate of, e.g., 10 Hz; and

- processes whose busy conditions are run during the time remaining from the operation of processes of all other priorities.

Four real–time process interaction forms are introduced as a job transmission. The process interaction form is defined in each query–receiving program of a process: for letters, for visits, for an order to execute a subprocess, and for requests for a certain resource.

In job–assigning operators, the name of the program receiving the job at the receiving process and, if necessary, the name of the accompanying message, is indicated. The program receiving letters, visits, and orders are always enabled for execution regardless of the receiving process status. The process interaction in the form of visits and orders is called an opened rendezvous.

In job receiving programs (for visits and orders) the course of the receiving process can be changed; in so doing, all the conditions for enabling such a change must be analyzed and taken into account.

Execution of the programs receiving jobs from higher priority processes is organized as a general queue operated upon during the periods between the scheduled realizations of busy conditions of processes at a given priority level.

Execution of programs receiving jobs from equal or higher priority processes is organized directly in the body of the job–generating operator, but with the priority of the receiving process.

Transmission of letter jobs is the simplest process interaction form. The letter sender process does not wait for a response about the execution of this letter's receiving program. Letters are necessary for transmitting the accumulating messages to lower priority processes.

Process interaction in the form of visits is characterized in that the process which generated the job continues only after the execution of the corresponding receiving program. A response message about the results of receiving a given visit is sent as a data word. Visits can be used to inspect a process and obtain data on its status without affecting its course; it is possible to organize the transmission of a certain conditional or unconditional job affecting the course of the process; it is possible to organize a shutdown, freezing, and subsequent recovery of the process execution, etc.

The generation of an order to execute a certain subprocess is similar to executing visits. Yet, in this case, the process which generated the order (immediately or following a number of conditions) goes into a wait state until the end of a given job's execution. Upon executing the job, the receiving process generates a response about the subprocess's completion.

A query form is introduced for organizing queries for a certain resource. The query execution is similar to visits but differs from them in that the query receiving programs may

be used only with permission from the receiving process. The incoming queries queue up for its query receiving program. The process generating the query goes into a wait state until its query is executed.

The organization of job receiving programs with a clearly defined process interaction form directly in each process makes it possible to shelter a given process from non-executed jobs and queries and insure an off-line development and debugging of process programs.

Let us examine the condition's program actions. Let us consider several possible condition operation phases.

The condition's primary phase is to process or convert input data entering during the current period and (if necessary) produce a certain data form which makes it possible to execute the requisite logical operations or branches.

Actual analysis of data received and processing with subsequent decision-making are performed during the second phase.

During the third phase of the condition, actions are carried out according to the decision made. Thus, e.g., if no change in condition is expected, these actions will consist in maintaining the necessary characteristics of output actions. If, however, a decision to branch to a new situation has been made, actions to disable the input actions related to the terminating situation will be additionally taken. These actions may already be classified as the next condition since they are a part of the process of determining its initial values. Source data on the commencing condition's actions will be subsequently ascertained; if necessary, the first execution of a portion of the branches and actions of the new condition may be performed. A given period terminates with an actual designation of a new condition of a given process for subsequent periods. We may also add that a number of actions in a situation may be unconditional and executed at any point in the situation, e.g., together with output branches.

After a new condition has been assigned, the first period of its execution does not differ from the subsequent ones at all, since all the initial values are established during the transient period and are related to the decisions made earlier.

In a general case, a condition may contain several input data processing phases, decision-making, and subsequent actions, although only the latest actions and the condition itself as a whole may terminate with assigning a new condition.

A condition structure in the form of a program consisting of two parts is introduced. The first part – single execution – is called the initial subsituation program. The second, executed at a fixed rate, is called the repeated part of the situation. Moreover, the structure of the repeated situation part is freely shaped by the user according to the problem to be solved. The POVTOR operator assigns the constraints for the condition operation during the current period.

The shaping of joint actions to activate or deactivate a situation for an entire group (block) of conditions which are assigned in the initial base condition of this block and the concluding program to deactivate the program execution of the situation block is additionally defined. This makes it possible to execute common deactivation actions during any process shutdown, and to execute common process activation actions during the start and recovery of an interrupted situation operation of the block.

We should note that only the subsituation alone can contain operators generating visit and order jobs. If the subsituation contains a wait operator for the end of an ongoing (in the form of an order) subprocess, the process will be delayed in this subsituation or, to be more precise, on this operator, until the end of the subprocess.

It is recommended that only one job be assigned from each subsituation. The situation from whose subsituation the job is assigned must not contain a recurrent part. Queries to transfer a unit (mechanism) to a new state are issued as jobs from processes which thus execute the management function for a group of units or the entire product as a whole.

Allowing for the set of unit statuses during its process, it is necessary to develop a set of programs receiving the corresponding jobs for transferring the unit to these states.

An analysis of mutual constraint versions of the process conditions reveals that given a process error or unscheduled shutdown, the process cannot always be restarted from the interrupted condition. For example, if a certain canceled motion of a mechanism consisted of several related situations (accelerating, maintaining speed, and braking), the interrupted motion must start from the original situation (acceleration).

With respect to the foregoing, let us clarify the concept of base situation in the situation block. The base situation in the block is understood as a situation that starts a certain complex multi–situation action. If there is an unscheduled shutdown, subsequent recovery of the process must be attempted from the base situation. Entrance to the situation block is permitted only through the base situation.

Let us consider the issue of shutting down a process in an emergency or other condition. The shutdown initiative does not originate from the process itself; if this is the case, it is not always known in which situation shutdown is necessary. In base situations of the situation block it is suggested that situations be assigned and a transition to them be made according to the query operator for a process shutdown.

Subsequently, if the VOSSTANOV operator is executed in a certain job–receiving program of a given process started from another process, a transition to the start of the interrupted situation block will occur in the given process.

Let us define the start of the situation block deactivation actions by the OTBOY operator and position them at the end of the situation block. Thus, given an OSTANOV operator, first actions will be taken according to the OTBOY programs, then branch to a situation defined in the base upon executing the process shutdown.

The situation block is also used to freeze the process operation in which given a STOPOR operator (from the job-receiving program), the program of the current situation block is run to its completion, the OTBOY program is executed, and the process is delayed until the VOSSTANOV operator is executed.

The introduction of the concept of "block" is due to the following structural constraints: a block program jump beyond the block limits is possible only up to the start of a certain other situation block.

In modern shop and machine tool control systems, the graphic display of the forms in which processes occur becomes especially important. This can be graphically illustrated by the GRAFCET system made by the *TELEMECANIQUE* Company.

In our case, it is impossible to borrow the aforementioned graphic systems directly for the following reasons:

- parallel control branches are realized in the system as individual parallel processes;
- a situation structure may contain actions taken before analyzing branch conditions;
- processes are executed at a different rate; and
- data are transmitted during the process interaction.

A graphic representation of the process

```
%%example;

inputs;
W,WX2(VB2,VB3);—names of word,its bytes,names of digits
'edgel'left'right4'far'close'norm'tilt1'tilt2;
'reverse'bottom'reverse'fromleft'fromright'pressure'work'nonstop;
W,real,limit,range1,initial,angle

outputs;
W,W14(BV4,BV5);
'control'swing'load'dleft'dright'round'dup'ddown'dtilt'slope'sign.6'bent'
zbent'vburn'screwin'return
W,speed,dose,recount,revolutions,accelerate,pressure,screwin,load

            —example of one process
[:]  process:14.overload;
read=close,norm,angle,range1,work,limit,
        initial,real,nonstop,far;
right=pressure,speed,acceleration,revolutions,recount,dose,screwin,
        sign.6,load;
```

```
      [#]             for order.from self;
                            if close*norm;then to.B2;otherwise W0=4 end;
      [#]             for order.to self;
                            if far*norm;then to.B3;otherwise W0=4 end;
      [#]             for visit.stopoverload;
                            if work;then end;otherwise shutdown;
      [#]             for visit.restoreoverload;
                            if nonstop;then end;otherwise shutdown;

                                  —example of only one situation block
B2    [B]             block.2?S12;T>20?S12;
       !                          —input situation
S6    [*]             sit.6;press=1;speed=10;accel=10;
      +—+                   if <angle>=20;then to.S10;otherwise to.S8;
S8    [*]   !         sit.8; revs>=range1?S10;
      +—+                         —conditional branch
S18   [*]   !         sit.18;text.B.topanel="error B",dec.position;
       !    !                     —message generation
S9    [=]   !         sit.9;panel.topanel;to.S16;
       !    !                     —generating visit with "topanel" message
S10    !   [=]        sit.10;spindle.acceleration;
       !    !                     —subprocess order
S11    !   [*]        sit.11;revolutions=initial+40//2+limit;conversion=W0/=0?S14;
       !   +—+              sign.6=close*norm+(press>6);to.S12;
S12    !   [*]  !     sit.12;dose=20;screwin=1;
       !   +—+              cycle:T>20?S14;real>20?S13;repeat;—repeated
S13    !   [?]  !     sit.13;wait.accelerate;to.S16
       !    !   !                 —waiting for acceleration completion
S14    !    !  [*]    sit.14;
       !    !   !     cycle:T>480?S15;repeat;—periodically repeated
S15    !    !  [?]    sit.15;wait.accelerate;to.S16;
       !    !   !                 —waiting for acceleration completion
S16   +—+—     [*]    sit.16;load=0;press=0 to.S6;

               [/]    clear;load=0;press=0;end;
```

In the graphic display system of the process structure referred to as GRAFPROTsESS, situations are designated by rectangles (as in the GRAFCET language), while process development paths – by communication lines with referencing directions from top to bottom and from left to right (see illustration). The situation's number is entered to the left of its rectangle.

In order to improve the display of reference data, comments, and branch addresses of all situations, a rule is introduced whereby all situations are displayed (as in the body of the program) one beneath the other. In so doing, the situation to which there is a transition from the previous situation is displayed immediately beneath the previous situation, while

if there is no branch, the following situation is positioned beneath it to the right, starting at the next vertical column.

The graphic display of a process's situation block contains its GRAFPROTsESS and immediately to the right of it – the situation program text or generalized data on the situation and situation numbers of other branches of the process which are not reflected in the displayed scheme. This makes it possible to relieve the GRAFPROTsESS of complicated branches between the situations permitted in the realization under study.

Busy process situations are shown by a special indicator in the GRAFPROTsESS.

The graphic resources described here are shaped on screen by the system using the character program text, which distinguishes it from the GRAFCET method where the graphic diagram is created interactively by the programmer.

In the method described here, system design may start with a graphic depiction of the GRAFPROTsESS using a pattern. Yet, after the program text has been entered in this GRAFPROTsESS, it is no longer necessary to form a picture which is immediately displayed on screen by the system and makes it possible to check the process program execution visually.

The display menu system is designed in such a way that by moving the cursor from one GRAFPROTsESS situation to the next, one can change (highlight) its displayed parts, go from a display of a group of processes to one process, from the GRAFPROTsESS to the program text of its process and vice versa, shift to displaying the retrieved subprocess or to a generalized diagram of a situation block interrelation of the process.

The computation core of the language under study is based on microprocessor device commands. This fact makes it possible to solve arithmetic and logical problems of the actual system interface operation efficiently. Despite the apparent similarity to high-level languages, there is a considerable difference: a source operand and a recipient operand are defined for all binary operations. The recipient operand can be the same for a number of consecutive operations and is positioned to the left of the operation sign.

The diversity in the coding and data processing forms leads to a situation where they acquire both special reserved and descriptive semantic user names in the system. The reserved names, in turn, may have an ordinal number or semantic descriptive forms.

Moreover, semantic names may be assigned not only to groups (chains or words) but to their individual parts (bits or word bytes).

Logical branching in the system is assigned by the IF... THEN... OTHERWISE... structure.

The language also provides for an individual conditional branch form whereby the branch line number of the process is indicated following a question mark at the end of the individual condition operation.

The programmer can compare the logical signal status (in an existing situation) to the value of the constant (0, 1) or logical expression. The expression (constant) is denoted by an equal sign following the modified signal name.

Conditional signal scanning by one is assigned by the signal name, while in the case of a zero scanning, a slash is entered in front of the signal name. Operations in the form of operations with operands (recipient and source) are defined in the language: scanning (′, /′) and toggling (#, /#) of the recipient digits over the source mask, comparing (==, /=, >, />, <, /<) operands, transferring the source value to the recipient, shifting the digits (❯, ❮) of the recipient by the source value, and arithmetic operations (+, –, *, //) where the results are entered into the recipient.

Operands may be defined by the word (byte) name of the register cell, indirect address in the listing, general purpose register name, word name of the process timer register, word name of the process status register, etc. Moreover, the source operand can be represented as a constant (octal, decimal, or binary–decimal number, or a mask where one highlighted digit number is given).

The fixed periodic nature of process execution at all control layers makes it possible to match the countdown operations with the execution rate of busy process conditions.

In essence, the timer operator is a condition (operand) in logical expressions or a simple branching condition in the program, and is realized as a subtraction of one (to zero) from the given process's system counter. The subtraction is performed each time when the timer operator is executed in a busy condition of a given process.

Let us consider the purpose and use of data chains. Data chains are one of the structural forms for creating and transmitting messages contained in the process interaction operators. The job being generated may contain the character text of the console instruction received by the process from the operator, a character text submitted for execution to another process controlling the technological program, a character text of the message sent to a display or another process, etc.

The data chain text is generated by the TEKST command which is followed by the chain type (*B* if it consists of bytes and *W* – if it consists of words), the data text chain name, and an equal sign, then a list of character data, constants, data formed according to the processing of the content of given memory registers, or other data chain elements according to its name.

The TEKST command makes it possible to create, merge, and transmit various data chains from one to the next.

The received text may be transmitted to other processes, other device's processes, to the operator console, etc. After transmitting a data chain to another process, the sending process loses its access to it.

In addition to generating and transmitting data chains, processes decoding the incoming data chains must be organized in machine tool automation systems control. To this end, an ANALIZ command which operates the search in the received chains of given (unknown) structures is introduced, i.e., it sorts out the received data.

The program controlling the entity consists of two principal sections. The first section contains data on the memory allocation and system peripherals, as well as identifies the names of input, output, and internal signals and data. The second program section consists of individual control process programs.

Each control process starts with determining its number (which is compared to the repetition rate of its busy situations) and the process name. The process start identification is followed by listing signal names and data which may be changed or scanned from the process (WRITE) and signals and data which may only be scanned from the process (READ). This is followed by all programs receiving the jobs entering a given process. After the job-receiving programs, process situation programs grouped into situation blocks are determined.

The software of the language under consideration are designed for developing, compiling, debugging, and starting programs controlling the electronic automation of complicated machine tools with single-processor ChPU devices, as well as for defining off-line and embedded control systems of manipulators, tests mockups, transfer head groups, retooled lines, etc.

Furthermore, the language's debugging resources enable the developer to: visualize messages from the electronic automation system and compiler and visualize debugger console commands; dynamically display electronic automation signal groups and a list of the chain of the most recent redesignation of busy conditions among processes and display text of programs controlling electronic automation equipment and their machine codes. During the debugging refinement of electronic automation equipment control programs, all permitted jobs and input signals are simulated from the control device panel. It is possible to refine programs and start the scheduled process operation period using instructions from the control panel rather than interrupts, i.e., step-by-step refinement.

In our opinion, the most important advantages of the FOKON language are that it is easy to visualize, compact, multifunctional, easy to program and debug, and has a high processor time utilization factor.

UDC 519.711.2:658.512.011.56

Automation of Decision-Making to Neutralize the Consequence of Unscheduled Situations in Flexible Manufacturing Systems

18610428D Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 6, Nov-Dec 88 (manuscript received 10 Feb 87; revised version received 27 Oct 87) pp 100-103

[Article by N. V. Globa]

[Text] A flexible manufacturing system (GPS) can be regarded as a certain deterministic system in which all processes of parts conversion and material resource movement are uniquely determined by the controlling actions generated by the GPS automatic control system (ASU GPS) and a human operator. Obviously, various unscheduled situations related to equipment failure, shortages of blanks, semi-finished products, and tools in the warehouse, rejected products, rush orders, and other causes are possible in any engineering system, including the GPS. Should this happen, decisions must be made to recover the lost GPS functions. These decisions may be expressed in reallocating resources among production modules, organizing repairs and maintenance, changing the system operation mode, changing the priority of queries being filled, etc.

Thus, the system must respond to any unscheduled change in the GPS status with a reaction consisting of a set of controlling actions aimed at neutralizing the consequences of the unscheduled situation to the system during the shortest possible time with the smallest possible losses in GPS performance.

In relatively simple GPS where the number of production modules is measured in single-digit numbers both in serial or mass production, the role of LPR [expansion unknown] is usually assigned to a human dispatcher. The man himself evaluates the emerging situation, weighs different options, makes a decision (possibly with the help of an on-line scheduled planning system (OKP) or other software), and transmits a message to the on-line dispatcher control system (ASODU).

Yet, the GPS configuration complexity and the range of functions performed by it are constantly increasing, and in the near future, such an allocation of functions between man

and the ASU GPS will become impossible. This is due to the fact that first, decisions must be made on-line; second, man always introduces a certain element of subjectivity to the decision being made; and third, man is incapable of making a sufficiently accurate estimate of the decision quality – hence the need to automate decision making.

To this end, we propose that a decision-making system for neutralizing the consequence of unforeseen situations in GPS (ASPR GPS) be developed in the framework of ASU GPS in order to perform the following principal functions:

- recognizing the unscheduled situation;
- making a decision about the necessary organizational measures; and
- logging data on the unscheduled situation.

The entire range of possible unscheduled situations can be tentatively divided into three groups:

- a situation where perturbations during the production process are external to GPS (e.g., in the case of a directive to start an unscheduled rush order);

- disruptions caused by such internal GPS factors as equipment failure, shortages of blanks, semi-finished products, tools, etc.; and

- latent GPS operation disruptions manifested in a systematic deviation of reliability and performance indicators, equipment utilization factors, and interchangeability from given limits which can be revealed only by analyzing the GPS operation over an extended period of time.

According to such a classification, we must introduce three different ASPR modes with the following system operation triggering types:

- given external perturbations, the ASPR operation is triggered by an ASODU constantly monitoring the GPS performance; and

- given an accumulation of GPS functioning data sufficient for making decisions regarding the presence of an implicit irregular situation and processing its characteristics, the ASPR operation is triggered by the dispatcher.

In the first two modes, the moments of ASPR switching are not regulated and it operates in real time. In the third mode, the ASPR operation session is realized only at regulated time moments (once in six months, once a quarter, etc.) and requires considerable computational resources for statistical data processing and multi-criterion optimization. Consequently, it is expedient to conduct this session during GPS maintenance or repairs when there is no shortage of computational resources.

All of the aforementioned ASPR functions are realized in any of the above operating modes.

If the existence of an irregular situation is evident (the first and second modes) and execution of a scheduled task is threatened, the recognition problem is reduced to determining the situation type by the irregular situation classification input into the system, assigning a systems code to the situation, and filling a file with variables. This file is transmitted over a program interface to the OKP system or another ASU GPS subsystem for making the calculations necessary for the ASPR. The plan for searching for the necessary decision is automatically plotted on the basis of the situation's systems code.

If the existence of an irregular situation is not evident (the third ASPR mode), the recognition problem amounts not only to establishing the situation's location in the adopted classification, but establishing the fact itself. If this is the case, the decision is made in two stages: first, we establish the possible causes of the irregular situation, then formulate an efficient plan for eliminating these causes.

To illustrate this, let us consider the processing of statistical data on GPS equipment failures. First we compute the normalize reliability and efficiency indicators of the GPS (the survival probability, availability, gamma-resource, efficiency conservation factor, etc.) and compare them to the standards. The existence or absence of an irregular situation is established by the deviation character of the computed quantities. Then we analyze the causes of failure and identify those which decrease the GPS reliability and functional efficiency most severely. And finally, the system computes a rational sequence of measures for liquidating the identified causes of failure.

In order to realize said ASPR functions, it is necessary to develop efficient means of storage, access, and processing of data on the GPS structure, its functioning patterns, the GPS interaction with kindred subdivisions, GPS operating experience, etc. The large diversity and considerable volume of data on the subject matter stored poses the problem of representing this knowledge in ASPR.

This problem can be solved by building a special-purpose knowledge base employing a semantic network apparatus [1, 2].

The semantic network is intended for storing data on concepts, events, and GPS characteristics and is the means by which the system "understands" the incoming message.

Said semantic network is a directed graph with marked arcs whose nodes represent concepts and arcs – semantic relations between them. There are three types of concepts: objects, i.e., GPS components; object actions; and object properties.

The class of conceptual objects consists of the following sets:

- flexible production modules (GPM);
- vehicles;
- blanks;
- fixtures;

- tools;
- product batches;
- an automated blank and tool warehouse; and
- service staff.

The concepts/actions are represented in the network as nodes marked by the name of the action from which the arc originates corresponding to semantic inflection relations. Inflection relations determine the subject of a given action and the characteristics (properties) of the object, subject, and the action itself realized in this case.

The concepts/properties are parameters whose specific value paints a complete picture of the situation. For example, the concept/action "GPS Functioning" corresponds with such properties/characteristics as the start time, end time, equipment utilization factor, interchangeability factor, efficiency preservation factor, etc.

Semantic relationships connect the system's objects, their functions, operations, and properties into one entity, as well as establish hierarchical structural constraints within individual concepts, thus transforming the knowledge about the system into a semiotic GPS model. Linguistic, logic, set theory, and quantified relationships are employed in the network.

Linguistic relations define the GPS functioning process by formalizing the relationships among its elements. They represent the aforementioned inflection relations among action verbs and the content of these verbs' tense forms.

For example, the statement "the part may be present in an automated warehouse cell or at the GPM" may be represented in the semantic network with the help of the semantic relation "to be present" as shown in Fig. 1.

Four types of nodes are identified in the figure: object concepts, property concepts, action concepts, and connectives. To represent this statement, it is necessary to introduce such relations from the theory of sets as "part–whole" and "set–subset" (arcs with this index originate from the subset). These relations establish the hierarchy of the system's elements, thus determining the GPS architecture.

The functioning patterns of the GPS and its components are represented with the help of quantifying relations of existence and universality, as well as disjunction, negation, and implication connectives.

The most important aspect of constructing an ASPR is to develop efficient decision search algorithms for semantic networks.

To search for the solution at each system layer, theorems are formulated and proven by the sequence method. Sequential calculus has a number of advantages [3] over the popular resolution method. By using the former, we can shorten the solution search time since

the sequential method makes it possible to link the theorem proof formalism to heuristic decision search strategies.

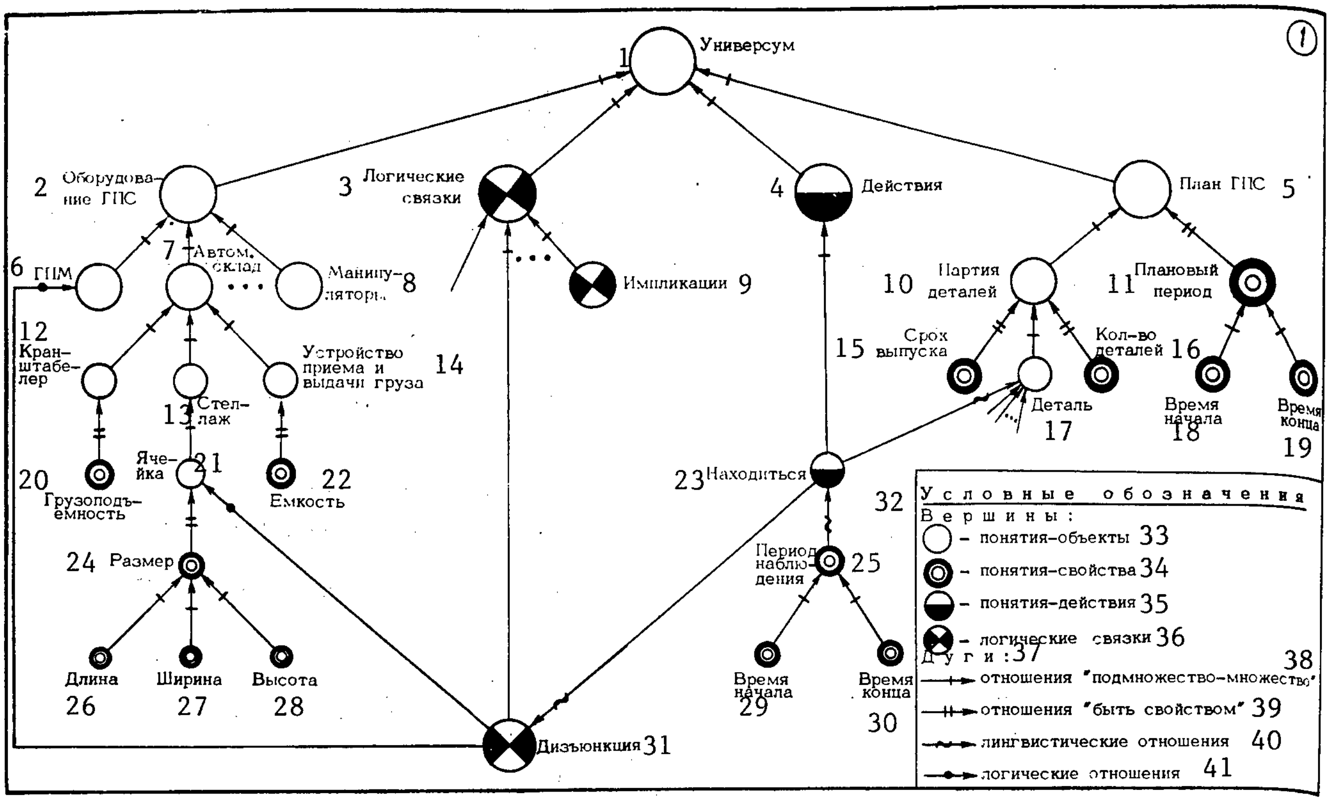


Fig. 1. Semantic network fragment of the GPS knowledge base: 1 – universe; 2 – GPS equipment; 3 – logical connectives; 4 – actions; 5 – GPS plan; 6 – GPM; 7 – automated warehouse; 8 – manipulators; 9 – implications; 10 – batch of parts; 11 – scheduled period; 12 – stacking crane; 13 – shelves; 14 – cargo handler; 15 – production date; 16 – number of parts; 17 – part; 18 – start time; 19 – end time; 20 – lifting capacity; 21 – cell; 22 – capacity; 23 – to be present; 24 – size; 25 – observation period; 26 – length; 27 – breadth; 28 – height; 29 – start time; 30 – end time; 31 – disjunction; 32 – legend, nodes: 33 – object concepts; 34 – property concepts; 35 – action concepts; 36 – logical connectives; 37 – arcs; 38 – "subset–set" relations; 39 – "be a property" relation; 40 – linguistic relations; 41 – logical relations.

For each type of problem corresponding to the above ASPR operation sessions, the problems are decomposed, as a result of which sets of problem layers are obtained. Each layer corresponds to a stratum of manipulated variables which are an array of parameters computed in the course of solving the subproblem and the source data for the lower-level problem layer [4].

In a general case, the statement of the problem of making a decision to neutralize the consequences of an irregular situation looks as follows.

Let there be a vector $X = \langle x_1, x_2, ..., x_n \rangle$ of the irregular situation's characteristics, as well as a set of uncertainties $W = \{\omega_1, \omega_2, ..., \omega_r\}$ expressing the existence of qualitative, non-formalized factors, conditions, and constraints of GPS and its elements.

Let us introduce an integrated GPS performance indicator:

$$G = \sum_{i=1}^{m} \alpha_i \frac{f_i^u - f_i^p}{f_i^u}, \quad \sum_{i=1}^{m} \alpha_i = 1, \quad \alpha_i \geqslant 0,$$

where f^u_i and f^p_i is the unit measure of performance of ideal and real GPS functioning, respectively; and α_i are the weight coefficients determining the "importance" of unit performance measures.

Each perturbation $\langle X, W \rangle$ can be placed in correspondence with a set of control actions $U = \{u_1, u_2, ..., u_p\}$. In other words, there exists such a finite algorithm A that $A: \langle X, W \rangle \to U$.

In turn, any elementary control action u_j transforms the input characteristic G of the manufacturing process, so that the final algorithm B of such a transformation is $B: U \to G$.

We shall assume each $\langle X, W \rangle$ perturbation to be random. Then the quantity G is also random.

The task of neutralizing the consequence of an irregular situation amounts to searching for a control action stabilizing the manufacturing process, i.e., resulting in a minimum variance of the quantity G during the time interval τ not exceeding a given time slice $[t_1, t_2]$, given a known distribution function $F(G, t)$:

$$D(G) = \iint_{-\infty}^{\infty} G^2 F(G, t)\, dG dt \to \min_{\tau \in [t_1, t_2]}.$$

The realization of the tasks thus formulated depends on the content of the set W. If all the parameters in the problem are defined, the problem is solved by multi–criterion optimization algorithms. If multi–criterion optimization algorithms cannot be used due to a large number of uncertainties, the solution is sought in the space of hypotheses by proving the corresponding theorems. In the remaining cases, the heuristic search for the solution is combined either with the sequence method or with multi–criterion optimization.

The results of solving lower–layer problems are checked in a GPS simulation in which the following principal problems are solved:

- designing a simulation experiment in order to obtain the maximum volume of data necessary for making a specific decision with a minimum amount of processing;
- conducting the simulation experiment; and

• processing the resulting outcome.

After simulating the GPS functioning, the decision is formulated in a form required by the user.

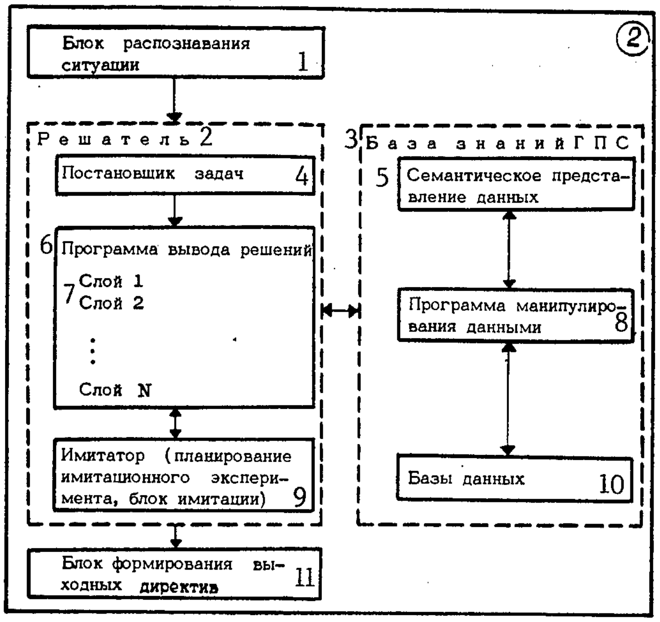


Fig. 2. ASPR GPS structure: 1 – situation recognition unit; 2 – solver; 3 – GPS knowledge base; 4 – problem definer; 5 – semantic data representation; 6 – decision–making program; 7 – layer; 8 – data manipulation program; 9 – simulator (simulation experiment design and simulator unit); 10 – databases; 11 – output directive formation unit.

Based on the aforesaid ASPR management principles, an ASPR structure shown in Fig. 2 can be offered.

The availability of a GPS simulation and deductive decision–making rules makes it possible to simulate not only the GPS, but the thought logic in solving organizational tasks.

The specific features of a concrete GPS are reflected in its knowledge base. The solver combines software systems and simulates the solution search logic ignoring the specifics of a concrete GPS, making it possible to develop unified software for the decision–making tasks in GPS for various purposes.

BIBLIOGRAPHY

1. Tyugu, E. Kh. "Kontseptualnoye programmirovaniye" [Conceptual programming]. Moscow: Nauka, 1984, 256 pp.

2. Popov, E. V. "Obshcheniye s EVM na yestestvennom yazyke" [Communicating with a computer in a natural language]. Moscow: Nauka, 1982, 360 pp.

3. Yefimov, Ye. I. "Reshateli intellektualnykh zadach" [Intellectual problem solvers]. Novosibirsk: Nauka, 1982, 320 pp.

4. Meltser, M. I. "Dialogovoye upravleniye proizvodstvom" [Interactive production control]. Moscow: Finansy i statistika, 1983, 240 pp.

UDC 681.324

Industrial Computer Network - A Systems Engineering and Process Functioning Base of the Integrated Automated Control System

18610428E Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 6, Nov-Dec 88 (manuscript received 16 Apr 87; revised version received 02 Jun 88) pp 104-106

[Article by A. A. Morozov, Z. M. Aselderov, V. I. Vyun, and A. A. Kupriyanov]

[Text] Based on the work of designing the start-up phase of the Integrated Automated Control System (KASU) of a large mechanical engineering enterprise, the authors of this article describe the software and hardware aspects of developing and designing industrial KASU computer networks [1].

The almost total identity of the production process structure of an overwhelming number of large enterprises and associations of the mechanical engineering industry and the need to integrate production processes and management activity at all levels necessitate the development of a problem-oriented local computer system (LVS) at each individual plant, design office (KB), or plant shop.

Let us call the integration of problem-oriented LVS distributed over the enterprise site into a single computer network insuring efficient functioning of all KASU subsystems an industrial computer network (PVS).

In designing a PVS, it is necessary to develop the following:

- PVS architecture and functioning principles;
- PVS hardware for various purposes, e.g., for computations, for matching heterogeneous network components, for displaying public and private access data, etc.;
- base network software;
- advanced programming methods and efficient testing means;

- an automated tutorial system to program application software packages in a network environment;

- database management control software, including that for individual databases (IBD), problem–oriented databases (POBD), and distributed databases (RIBD); and

- problem–oriented LVS, e.g., management (situation) control systems for making decisions in conflicting situations.

General Requirements for PVS and its Components. The experience of developing and analyzing the tasks which must be solved by the enterprise KASU, as well as the multilayer and multiaspect control structure of the production process and design and process tooling for manufacturing, enables us to formulate the following PVS requirements:

- virtualization of network resources, i.e., a unified machine, terminal, process, data, and program identification (addressing) system in a single user job control language in a network environment independent of either the computer itself or its operating system, must be realized for all computer classes (types and generations), as well as a unified system of control protocols for receiving, transmitting, storing, and processing data in any form (text, speech, graphics, and images of moving and stationary objects);

- insuring the interaction of each pair of both computers and terminals with equal rights to organize data interchange relative to each other;

- availability of public access resources necessary for developing economic and mathematical models, data and knowledge bases, and information retrieval and reference services;

- availability of a means of protection from unauthorized users at all data processing, storage, transmission, and display stages; and

- availability of a specialized administrative network service.

PVS Architecture. Said functional requirements and basic principles and criteria for developing a KASU with its multilayer control hierarchy uniquely define the PVS architecture concept.

According to this concept, a PVS must be a multilayer hierarchical system of interacting functionally–oriented LVS.

LVS are standard architectural components of PVS and are, in turn, constructed on a hierarchical modular principle, making it possible to increase output by varying the number of machine and processors and replacing them with more powerful ones, increase the capacity of computer RAM and external memory, and increase the number of peripherals and replace them with new ones.

In the proposed approach, the base PVS software (BSPO) is a set of routines which are "embedded" in each other and realize network services: data transmission, transport, and user support.

The network's data transmission service corresponds to the first two layers of the VOS-MOS model [2–4] and insures both the logic channel control and access to physical connection hardware, as well as procedural resources of data transmission through a physical medium.

The network's transport service (TSS) establishes and disconnects calls and controls the routing, transmission, and stream of data. The TSS incorporates the network's transmission service.

The network's user service corresponds to all layers of the VOS–MOS [expansion unknown] model and contains the entire set of resources for meeting a random user's needs. Here, a network subscriber is understood as a random program (application program, operating system (OS) program module, translator, loader, database management system (SUBD), etc.) having their own network address.

The network's user service is divided into sets of functionally–oriented program systems:

- servicing application programs (text and image input and processing systems, SUBD, terminal and interaction control systems, file transmission and management systems, ARM [expansion unknown] servicing, etc.); and
- management support (operator control systems, situation control systems in operation rooms, systems tracking the network component operation, collecting and providing data, diagnosing, restoring serviceability, controlling configuration, etc.).

In developing BSPO, it is expedient to select from the entire set of the user service's program modules a subset of functionally independent program modules which are common for a number of application level protocols in the VOS–MOS model and, together with TSS macros, serve as the basis for developing a unified virtual language for remote user job control.

Start–Up PVS Complex of a Mechanical Engineering Enterprise. The above requirements for PVS were taken into account in designing a start–up mechanical engineering enterprise system.

The start–up complex architecture is shown on page 3 of the cover.

The PVS contains the following types of machines: a YeS computer (models YeS1033, YeS1045, YeS1055M, and YeS1060) and an SM computer (models SM–4, SM–1420, and SM–1800).

The start–up system's data transmission network includes:
1) off–the–shelf resources of YeS and SM multicomputer operation:

- computer communication devices (A71119),
- "channel–to–channel" adapters (YeS4060),
- interprocessor communication adapters (SM–4503),
- asynchronous data transmission multiplexers (SM–8514),
- remote communication adapters (SM–8502), and
- terminal processors for the BARS [expansion unknown] system (YeS2602M);

2) experimental prototypes of YeS and SM multicomputer operation resources:

- sets of remote intercomputer communication (KDMS), making it possible to combine up to five machines (YeS and/or SM) located up to 1,000 m from each other, given a transmission rate of 48 Kbit/s (over a telephone channel);

- local intercomputer communication systems (KLMS), making it possible to combine up to four YeS computers with a maximum distance of 600 m between them, given a transmission rate of 700 Kbit/s (over a multiple conductor coaxial cable); and

- a base local data network (BKILS), making it possible to combine up to 50 machines (YeS and/or SM) at a distance of up to 200 m, given a 2 Mbit/s transmission rate (over a single conductor coaxial cable).

The start–up complex PVS is a multilayer hierarchy system of interacting LVS.

The first layer is the LVS at a head data center (GIVTs) helping to solve all functional tasks of the production association's integrated ASU and organizing the collection, processing, and transmission of technical and economic data for production association–level control.

The second layer are LVS at public data centers (IVTs) servicing the needs of plants, industries, and design offices (blank forging, mechanized assembly, transfer assembly, metallurgical, logistical support control, and production design and process tooling) which employ the territorial data storage and processing principle.

The third layer are control centers, i.e., shop–level LVS which are functionally oriented at controlling warehouses, machine tool groups, etc.

LVS at GIVTs and IVTs and shop–level LVS, in turn, are based on the hierarchy principle: the first layer is based on YeS computer systems, the second – on SM, and the third – on the basis of microcomputers and professional PCs (at the development stage).

The PVS software includes a number of routines; their components are distributed among the network's nodes.

1. A TSS [5, 6] represents an interrelated complex of program modules present in the network's operating computers and transport stations (TS) and insures data interchange among user processes of transport service users. The TSS is set up as a hierarchy of three interacting functional layers at which the following tasks are executed: control of data transmission over communication lines, control of data transmission in transit, and control of through data transmission.

The software architecture of the network's transport service meets the requirements of layers 2–4 of the benchmark VOS–MOS network model. The following serve as transport service modules in each computer of the network: a network process monitor (SMP) executed as YeS, SM–4, and SM–1420; and SM–1810 realizing a TS as software.

Two application processes can be connected by a virtual circuit through which their interaction is regulated by the transport protocol. The protocol is understood as a set of rules of logic and timing of the data interchange between two processes, as well as an agreement governing the format and semantics of these data. The protocol unit hierarchy of TSS data is represented by a frame, a packet, a datagram, or a letter.

User processes transmit to each other data files of various length stored in their RAM through logic I/O points which are defined beforehand in TSS. Let us refer to the data unit transmitted by the TSS during one access to it as a letter. The maximum letter size is determined by the parameters of the protocols used in TSS. In this particular implementation, it does not exceed 27,648 bytes. When transmitting a letter between user processes, the TSS does not process the letter's content.

A letter may contain any sequence of bits, characters, and codes. To identify ports, i.e., logical I/O points, unambiguously, a unique network port number is assigned to them for the duration of the user process, enabling the TSS to distinguish ports from one another.

The port's network number has a hierarchical structure and is a concatenation of the following elements: the network computer's number and the port's local number in a given computer. The local port number also consists of two parts: the TSS user number and the port number in the application process. This makes it possible to reserve to the latter a certain space of network port numbers and use it at one's discretion.

2. An information relationship monitor (MIS) [7] of dissimilar SUBD is intended for developing a distributed database. The MIS insures the data interaction among the databases maintained by the SPEKTR and INES SUBD at YeS computer network nodes and MIRIS and FOBRIN SUBD at SM computer network nodes. In so doing, the programmer, i.e., the user program developer, may not even know what and where data are located and by what SUBD resources they are maintained. A service rendered to MIS by TSS is used as the transmission medium. The RAM volume for the MIS's resident part does not exceed 200 Kbytes for YeS computers and 40 Kbytes for SM–4 computers.

3. A remote job input/output (UVVZ) realizes certain functions of the remote job input protocol and is intended for processing user commands in controlling the execution of jobs

distributed over YeS computer–based network nodes. UVVZ services can be accessed from the terminals supported by the "Vektor" system.

The UVVZ insures the following:

- controls the processing of data distributed in the network (tuning to a job library, reading and writing, viewing, editing jobs, queuing for processing, displaying the status and messages, and changing the job status); and

- supporting the functions of the systems and problem programmer, which are executed by the commands controlling OS objects in the network (instruments, directories, lists, queues, OS resources, volumes, and messages).

A service rendered to UVVZ by the network's TSS is used as the transmitting medium. The RAM (OP) volume necessary for UVVZ functioning is no more than 150 Kbytes.

4. A file transfer system (SPF) realizes the file protocol functions and is intended for transporting magnetic tape and disk files (data sets) and library sections in the network upon user request. The SPF consists of a combination of components distributed over YeS and SM computer–based networks.

The SPF enables the users to do the following:

- control the transmission, reception, creation, removal, and renaming of files;
- converting data into the receiving node's codes; and
- receiving data on the course and results of file transmission.

Summary. The above requirements imposed on PVS KASU and the design adapted in implementing the start–up complex constantly being improved and developed both with respect to their hardware (using microcomputers) and software components (subsequent protocol standardization, expanding service, etc.).

A more detailed description of certain BSPO components can be found in this issue of the journal and will also be published in one of the future issues.

BIBLIOGRAPHY

1. Skurikhin, V. I., Morozov, A. A. Integrated automated control systems. Features of architecture and development trends. USiM, 1976, No. 2, pp. 5–11.

2. Yakubaytis, E. A. "Arkhitektura vychislitelnykh setey" [Computer network architecture]. Moscow: Statistika, 1980. 279 pp.

3. Yakubaytis, E. A. "Lokalnyye informatsionno–vychislitelnyye seti" [Local computer networks]. Riga: Zinatne, 1985, 283 pp.

4. "Lokalnyye vychislitelnyye seti: opyt mezhdunarodnoy standartizatsii" [Local computer networks: International standardization experience]. Moscow: MTsNTI, 1984, No. 27, 120 pp.

5. Aselderov, Z. M., Konozenko, V. I., Pilipenko, V. G. Data transmission in an automated KASU network. In the book: "Voprosy konstruirovaniya i vnedreniya programmno–tekhnicheskikh kompleksov ASU" [Automatic control system software and hardware: Design and implementation]. Kiev, Cybernetics Institute at the Ukrainian Academy of Sciences, 1983, pp. 34–42.

6. Pilipenko, V. G. Data transmission channel control in distributed CAD. In the book: "Kompleksnyye avtomatizirovannyye sistemy upravleniya" [Integrated automated control systems]. Kiev, Cybernetics Institute at the Ukrainian Academy of Sciences, 1982, pp. 74–81.

7. Pilipenko, Yu. G. Software for data interaction in distributed databases. In the book: "Razrabotka i vnedreniye tsifrovykh vychislitelnykh complksov i sistem raspredelennoy obrabotki dannykh" [Design and implementation of digital computer systems and distributed data processing systems]. Kiev, Cybernetics Institute at the Ukrainian Academy of Sciences, 1986, pp. 82–87.

UDC 681.324

Base Local Computer Network Complex Hardware

18610428F Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 6, Nov-Dec 88 (manuscript received 16 Apr 87; revised version received 25 Jul 88) pp 107-108

[Article by Ye. P. Moiseyenko and G. I. Sinyayev]

[Text] The base local computer network complex (BK ILS) is one of the principal components of the industrial computer network (PVS) of a large mechanical engineering enterprise described in [1].

Communication between PVS computers providing a service corresponding to the transport layer of a benchmark open system interaction model recommended by the International Standards Organization [2] is realized on the basis of the BK ILS. In today's seven-layer model, BK ILS functions insuring communication with a user in the computer above and communication with the trunk cable below are distributed over the lower four functional layers (transport, network, channel, and physical) in such a way that logically independent functions can be developed and executed independently of each other. Access to these functions is also independent.

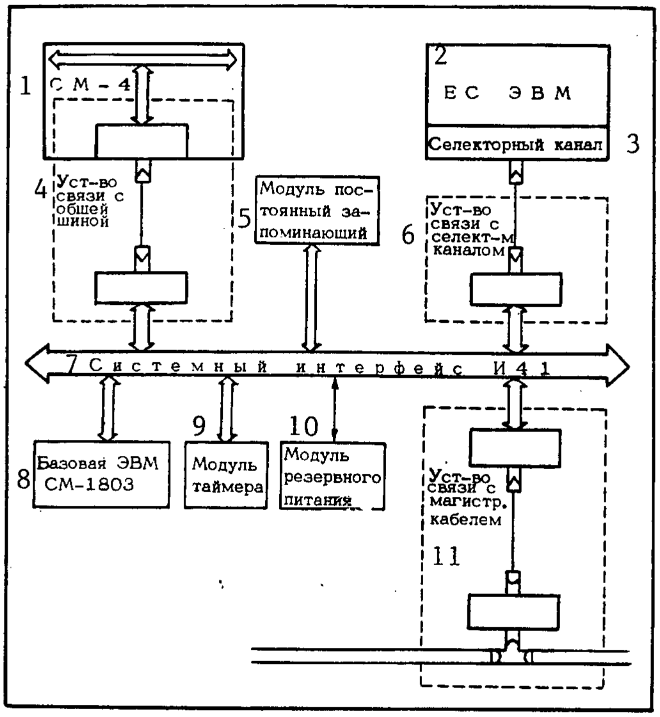
Such an approach made it possible to use the ideology of the following documents in developing BK ILS [2]:

- at a transport layer – the *ECMA–72* standard;
- at a channel layer – the *ECMA–82* standard; and
- at a physical layer – the *ECMA–81* standard.

Stations connected to a coaxial trunk cable are the elements of the BK ILS modular structure. A BK ILS–based transport station (TS) is executed as a base SM–1803 computer which is due primarily to the availability of the advanced I41 systems interface, the availability of modules and devices necessary for realizing a TS among SM–1800 components, and the availability of 17 free slots in the design for placing modules and devices.

Moreover, the base SM–1803 computer provides to all the modules and devices contained in the TS the necessary power supply and ventilation.

The number of TS depends on the number computers being integrated and their mutual position. A TS contains standard and optional sets of items (see figure).



1 – SM-4; 2 – YeS computer; 3 – selector channel; 4 – communication to UNIBUS; 5 – permanent storage module; 6 – communication with selector channel; 7 – I41 systems interface; 8 – base SM–1803 computer; 9 – timer module; 10 – standby power supply module; 11 – communication with trunk cable.

The base set [3] includes the following:

- a base SM–1803 computer;
- a permanent storage module;
- a timer module;
- a trunk cable communication device; and
- a standby power supply module.

The optional set consists of the devices for linking computers to TS: to communicate with the UNIBUS (USOSh) and to communicate with the YeS computer selector channel (USES).

The number and type of communication devices contained in TS depends on the number and type of computers connected to TS. Up to a maximum of four computers (either YeS or SM or both) can be connected to a TS.

The SM–4 computers are hardwired to the network through an off–the–shelf USOSh – the SM–1800.4502, while the YeS computer – through a device executing the commands of a regular YeS4060 computer. The transmission rate does not exceed 400 Kbyte/s in either case.

The trunk monochannel hardware – the device communicating with the trunk cable – is similar to the one in [4] and insures execution of physical and channel layer protocols.

Specifications

Transmission rate	4,096 Mbit/s
Channel operation mode	random multiple access with carrier control, conflict detection, and baseband transmission
Transmitting medium	coaxial cable
Maximum trunk cable length	no more than 2,000 m
Maximum station distance from trunk cable (transceiver)	no more than 200 m
Dual–input RAM capacity	32 Kbytes
Block length transmitted in one operation	(10–2,047) bytes
Error detection method	cyclic redundant code with an $X^{16} + X^{12} + X^5 + 1$ polynomial

The remaining part of the channel, network, and transport layer protocol is executed as software. At the TS itself, it may reside in RAM (OZU) or ROM (PZU) while in the computers linked to the network – in their RAM.

The principal BK ILS function is to render transport services to users. A user interacts with the network's transport service (TSS) with the help of named transport primitives. Each primitive has one or more parameters, and is linked to an event control unit. Transport primitives are generated both by users and the TSS and enable the users to do the following: connect or disconnect from the TSS; complete and disconnect a transport call; and receive and transmit data over a transport connection.

TSS users are represented by transport addresses. Each transport address uniquely defines users within the network. Data interchange between a pair transport addresses is independent of their location.

A transport address contains the computer number – the sole number in the network – and the port number – the sole number among a given user's ports. The maximum value of each number is 255. In addition to user port numbers, transport addresses may be assigned by the network's administrative service.

The user and TSS exchange transport primitives through ports. The total number of TSS ports accessible for simultaneous user connection is limited only by TS resources and is equal to 255 in this BK ILS version.

Within the BK ILS, the TSS is represented by identical transport modules positioned at TS. The TSS functioning, i.e., the interaction among transport modules, is determined by the transport protocol.

To exchange data between users, i.e., between two transport addresses (or between two ports), associations referred to as transport liaisons are established in TSS according to the transport protocol.

Transport modules interact by using a channel–layer service insuring a frame transmission from the TS buffer memory to the monochannel and its reception into the buffer memory from the monochannel. The frame is a unit of data interchange in the monochannel and has a structure shown below.

Feature	Number of bytes
Preamble	5
Recipient address	6
Sender address	6
Length	2
Data	10–2,047
Checksum	2

In packing the frame, all of its fields, with the exception of the preamble and the checksum, which are formed by the hardware, are filled.

Frames are sent to the channel serially, i.e., the successive frame transmission to the channel is enabled following the completion of the previous frame transmission. The frame transmission control whose functions also include error detection is interrupt–driven.

Routing in BK ILS amounts to comparing the computer number contained in the transport address to that of the TS connected to the desired computer.

The transport interface insures communication between the users and TSS, namely:

- recognizing transport primitives generated by users;
- forming a transport interface data block (BDTI) according to the transport primitive and its parameters;
- recognizing transport primitives generated by the TSS and transmitting them to the corresponding users;
- controlling the BDTI interchange between computers and TS, both on the computer and TS side; and
- simultaneously linking all users of a given computer to the transport service.

Transport primitives are defined by a set of macro definitions. The user dispatcher enables all users of a given computer to access a TS simultaneously. To this end he, on the hand, maintains communication with all the users desiring to use TSS services and, on the other, constantly maintains communication with the station through a common memory segment, in which two queues – an input and an output – are organized, by controlling access to TS.

Access to TS is controlled by duplex data interchange between the station and the computer and depends on the type of computer. A standard SM–1800.4502 device and an XB DRV driver in the OS RV system are used for connecting the station to an SM computer, while programs employing the BSAM access method expanded by the MMS multicomputer operation equipment are used for connecting to a YeS computer through a device executing YeS4060 commands.

The implementation simplicity of interlayer BK ILS interfaces made it easier to connect the lower two BK ILS functional layers (physical and channel) to the network by using another TSS – a "Communication Process Monitor" software package [5].

BIBLIOGRAPHY

1. Industrial Computer Network – A Systems Engineering and Process Functioning Base of the Integrated Automated Control System. By A. A. Morozov, V. I. Vyun, Z. M. Aselderov, A. A. Kupriyanov. USiM, 1988, No. 6, pp. 109–111 [sic].

2. Local computer networks: International standardization experience. In the book: "Metodicheskiye materialy po paketam prikladnykh programm" [Methodological materials on application software packages]. Moscow: MTsNTNI, 1984, No. 27, pp. 69–175.

3. "SM–1800: Arkhitektura, programmirovaniye, primeneniye" [SM–1800: Architecture, programming, and applications]. By A. V. Giglavyy, N. D. Kabanov, N. L. Prokhorov, A. N. Shkamarda. Moscow: Finansy i statistika, 1984, 136 pp.

4. Crane, R. Application software and controller for connecting DEC computers in the ETHERNET network. *ELEKTRONIKA*, 1981, No. 25, pp. 27–32.

5. Aselderov, Z. M., Pilipenko, V. G. Network software for a heterogeneous computer network. USiM, 1988, No. 6, pp. 114–117.

UDC 535:681.51

Information Retrieval System of Archival Space and Aerial Remote Sensing Data

18610428A Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 6, Nov-Dec 88 (manuscript received 04 Jan 87; revised version received 18 Jan 88) pp 90-91

[Article by Yu. V. Chukin]

[Text] Remote sensing plays an important role in the range of methods used for mineral exploration. In particular, they help to define the scope of geographic and geological prospecting more precisely, increase the reliability of maps, and lower the labor outlays and time for their compilation. In developing a remote sensing data (MDI) processing system, one of the most urgent tasks is the collection, storage, processing, and retrieval of large volumes of data – not only the images themselves and charts, but collateral data on their presence, location, and various characteristics, i.e., the task of database organization and management (IO) of these processes. Consequently, IO automation is among the principal tasks facing the MDI processing system. Moreover, since research in this field is complex, difficult, and informal, today the IO process rather than the research process itself is the subject of automation.

The need to automate IO is primarily dictated by the low productivity of the traditional "manual" IO methods, which already impedes the growing volume of data used in today's remote sensing. At present, the principal problem lies not in the lack of necessary data but in their timely availability, retrieval, and transmittal to users in the required form and format (thus, e.g., there are tens of thousands of mapping and hundreds of thousands of photographic documents in a standard regional MDI processing center).

In order to automate IO tasks, we must develop appropriate resources – an information retrieval system (IPS). Due to the complexity of the images used in remote sensing, the most appropriate method of retrieving the necessary images today is searching their identifiers and concomitant parameters. The IPS were developed to increase efficiency and speed up MDI processing as a whole. This is attained by the following:

- increasing the accessibility of a large volume of data on MDI;

- providing users with reliable on–line information whose content and scope insures that research and production tasks are accomplished well;

- increasing labor productivity of users and service staff by freeing them from the routine tasks of retrieving, preparing, and presenting the necessary materials; and

- increasing the throughput of the accounting record–keeping service.

A data retrieval system for space and aerial data (IPS AKSIOMA) developed for maintaining, storing, and retrieving data on aerial and space photography which are located at aerial and space video data processing centers and used in remote sensing is briefly described below. This IPS may be used as an off–line retrieval system or serve as a supporting storage and maintenance subsystem of computer–based archives of aerial and space data for a regional or branch–wide MDI processing system, either operational or under development, at a number of ministries and departments (Ministry of Petroleum Industry, Ministry of Geology, Agroindustrial Committee, etc.).

The goal of IPS is automatic execution of the following operations involved in solving research problems using remote sensing of the Earth's natural resources:

- multifunctional retrieval of data based on a set of criteria and conditions, enabling the user to retrieve on–line data directly in response to both standard and non–standard incoming queries;

- visual display of the retrieved pictures' location in a 1,000,000– or 200,000–line list format; and

- publishing secondary data sources – catalogs, indices, and subject lists which present various aspects of source data, in an optimal volume and in a user–friendly form.

General System Architecture. The AKSIOMA IPS is executed as an SM–4 computer with an RV [expansion unknown] operating system (OS). The operating system and external memory volume are 96 Kbyte and 30 Mbyte, respectively. A MIRIS database management system (SUBD) is used as the IPS kernel.

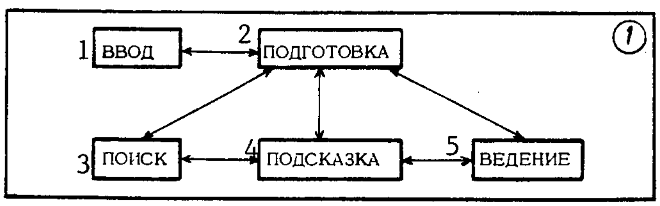


Fig. 1. Block–diagram of the IPS AKSIOMA: 1 – input; 2 – preparation; 3 – search; 4 – help; 5 – maintenance.

The AKSIOMA IPS consists of five subsystems; the rule governing the transitions among them is spelled out beforehand (Fig. 1). Each subsystem contains a corresponding set of instructions. The PRODGOTOVKA subsystem prepares the system for operation, per–

forms its on–line control, and selects the operating mode. The VVOD subsystem collects, prepares, and inputs source data into the system. The POISK subsystem contains instructions for selecting a database, responding to queries, processing the results, and providing them to users. The VEDENIYe subsystem corrects systems data, as well as produces various catalogs and subject lists. The PODSKAZKA subsystem supplies users with the necessary descriptions, instructions, and system operation help messages.

This system's software was developed so as to enable it to overlay programs in computer RAM. Overlay program loading structures are defined during the task compilation. To realize these possibilities, the software provides for a certain modularity. The system contains 33 modules written in the PASCAL language.

The system's database consists of three files: space pictures, aerial pictures, and space photomaps. A space picture description contains 23 attributes, eight of which (the scale, nomenclature sign, survey date, survey time, survey type, film number, frame number, and inventory number) may be used as search keys.

The manager and application programmer communicate with the system using the MIRIS SUBD language resources. A table–oriented, highly declarative query language has been developed for end–user communication with the system. When formulating queries in this language, the user only has to fill an eight–column table displayed on screen (search keys) with the necessary values. The language's simplicity and streamlined semantics enable the operator to think in terms of table names with which he can perform familiar highlighting and search operations.

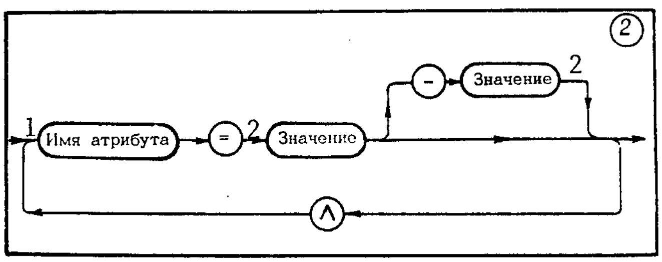


Fig. 2. AKSIOMA IPS query class: 1 – attribute name; 2 – value.

Today, the existing language version realizes the class of queries shown in Fig. 2. Data can be searched by any one or group of search keys. Each search key (except for the nomenclature sign) can be defined either by its precise value or by a range of values. Experience shows that the query class realized there is quite suitable for end users in a given subject area who, as we know, try to avoid compiling a complicated query at any cost, even if there is a universal language capable of formulating complicated queries, by substituting it with a series of simple queries.

Since spatial information as such is not stored in the system, conventional data management and search methods are used. The system is distinguished in that it has the resources to interpret specific applications terms and concepts (cartographic in this case), as well as to process and visually display the position of found pictures. Thus, the system

supports and recognizes standard nomenclature signs (numbering signs of pictures on various scales referenced to the graduation adopted in topography) which correspond to a scale from 1:1,000,000 to 1:10,000. Two types of searches by the nomenclature sign can be executed: a "downward search" and an "upward search".

During a downward search by a nomenclature sign defined by the user, the system produces all pictures on a scale from the one corresponding to the sign to 1:10,000, i.e., all pictures on a scale defined by the sign or smaller pertaining to said territory. During an upward search, the system produces all pictures on a scale from the one corresponding to the nomenclature sign to 1:1,000,000, i.e., all pictures on a scale defined in the sign or greater containing said territory. During the upward search, it is possible to indicate the geographic position of the point on the territory of interest to the user instead of the nomenclature sign. Pictures on the requisite scale are selected by an additional (simultaneous) instruction in the query.

The system is capable of displaying on an alphanumeric terminal the position of the retrieved pictures in the framework of a 1,000,000– and 200,000–line list. If proper hardware is available, the images can also be viewed.

Plans call for expanding today's level of the system described here along the following lines:

- Automate the selection and ordering of space pictures directly from catalogs using magnetic media, making it possible to bypass the manual data entry stage;

- complement data on MDI availability with data on how much is known about this territory (geological, geographic, etc.); and

- add to picture data a number of parameters describing their content (e.g., condition, weather conditions, certain indices, etc.), thus expanding retrieval capabilities.

Abstracts of Articles in MASHINOVEDENIYE, No 2, 1989

18610455 Moscow MASHINOVEDENIYE in Russian No 2, Mar-Apr 89 (manuscript received 24 Nov 89 [sic]) pp 109-112

[Text]

UDC 629.735.33.015.4.017

Underlying Principles of Reliability and Strength Standards in Mechanical Engineering. A. F. Selikhov, V. L. Raykher, and Yu. A. Stuchalkin, Moscow, pp 5-10

The principal strength and reliability features to be standardized are formulated. The most important and universal standardization principles related to the feasibility of ways to insure strength and reliability, the entity's quality requirements, the extent to which the factors resulting in failures are considered, operating conditions, and the presence of feedback are considered. The imporatnce of standardization as a system is stressed. Examples from the aviation industry are cited.

UDC 681.5.01

The Principle of Designing Automatic Control Systems With the Help of Partial and General Models. Yu. B. Podchufarov, Tula, pp 11-15

Given a robust design of engineering system, the outcome of selecting the constraints of its members and choosing the values of control paramters and algorithms obtained at the previous stage is an integral part of the results of all the subsequent design phases. The design process is robust if it is divided into stages from in situ to in toto. In so doing, problems at each phase are solved by the totality of requirements for the dynamics, accuracy, and reliability of engineering systems, not for the entire phase space but only for a certain area. The concepts of total and guaranteed inclusion of the results of the previous stage into the outcome of the subsequent design stage are introduced. An example of designing an electromechanical control system based on the proposed principle is cited. Figures 2; references 6.

UDC 539.4:620.193

Predicting the Propagation of Corrosion Fatigue Cracks. V. V. Bolotin, A. S. Ryabtsev, and A. S. Shubin, Moscow, pp 16-23

The corrosion fatigue model is refined by simultaneously considering the energy balance conditions in a system made up of an object with cracks and accumulated microdamage near the front of the cracks. Equations are derived for a quasi-static approximation in which the the crack propagation is approximated by continuous functions. The effect of the loading conditions and initial crack dimensions on the propagation, the accumulation of mechanical, corrosive, and total microdamage, and the change in effective curvature radii at the end of propagating cracks is analyzed. Figures 5; references 3.

UDC 629.4.023.11:539.4(045)

Improving the Strength Analysis of Crucial, Highly Loaded, Welded Locomotive Frames. B. B. Bunin, Ye. I. Zhuk, V. P. Kogayev, V. G. Perfilov, and T. M. Ponomareva, Kolomna and Moscow, pp 24-31

The results of a study of the loading condition of locomotive trucks during their operation are presented, the stress amplitude distribution laws truck frame elements are derived, the results of bench tests to determine the fatigue limit of full-scale side-frames are cited, and the probabilities of their fracture are found using the procedure developed at the State Mechanical Engineering Institute based on the linear and corrected fatigue damage accumulation hypotheses. The possibility of lowering the standard safety margin for these parts is demonstrated by comparing the results of the truck frame strength analysis obtained by various methods. Figures 2; tables 6; references 4.

UDC 539.442

Using the J-Integral Concept to Describe the Low-Cycle Failure. A. V. Getmanchuk, Yu. G. Dragunov, V. I. Yegorov, and N. D. Sobolev, pp 38-40

Low-cycle failure kinetics of 12Kh18N10T and 00Kh16N15M3B steel in tubular samples are examined from the viewpoint of non-linear failure mechanics. The J-integral is used as a criterion. Figures 2; references 8.

UDC 620.178.3

Fatigue Resistance of the VTZ-1 Alloy Due to Surface Hardening. M. N. Stepnov, M. G. Veytsman, B. V. Giatsintova, and L. V. Agamirova, Moscow, pp 41-44

The effect of surface hardening on the fatigue resistance of cylindrical girdle specimens from VTZ-1 alloy is examined. Stock samples were heat-treated

under three conditions, thus imparting various structure and mechanical properties in three batches. To study the hardening stability due to an extended static loading, cylindrical samples made from bar stock were divided into four groups. The first two gropus of samples, one of which had already been hardened, were exposed to static loading for a year. An analysis of the results of experimental investigations reveals that the efficacy of hardening samples from VTZ-1 titanium alloy depends on the microstructure character, and reaches peak values given an equiaxial and mixed structure. Cold-work hardening reduces the spreading of fatigue resistance. An extended exposure of hardened samples under a load did not change the efficacy of hardening. References 3.

UDC 621.822.6-19.001.24

Estimate of the Fatigue Durability of Rolling Bearings Under Random Loads. N. N. Dobromyslov, Moscow, pp 45-49

A method is proposed for calculating the failure-free operation and longevity and indices of rolling bearings under random loads. The method is based on kinetic damage accumulation models developed for a multi-cycle fatigue strength of structural parts and members. Here, these models are applied to contact fatigue. The effect of the actual radial clearance in the bearing on its durability is taken into account. The method makes it possible to take into account the variability of load on rolling bearings when specifying their useful life. Figures 3; references 4.

UDC 539.3

The Design of Sectional Composite Shell Structures Under Static and Dynamic Impacts. I. N. Preobrazhenskiy and V. G. Dmitriyev, Moscow, pp 50-55

A numerical method of analyzing sectional shell structures made from composite materials under the effect of static and dynamic loads is presented. The shell-type construction is formed by joining shells of revolution with the help of elastic rings. The original relationships, which take into account geometric non-linearity in a quadratic approximation, are based on both the Kirchhoff-Love and Timoshenko hypotheses. The finite difference and determination methods are used to solve the problem numerically. The non-stationary equation form adopted in the determination method makes it possible to construct a single difference procedure for solving both static and dynamic problems. The efficacy of the numerical method thus developed is demonstrated by a computational experiment. Figures 5; references 5.

UDC 621.891+539.538

A Generalized Model of External Friction and Wear. B. M. Silayev, Kuybyshev, pp 56-65

It is suggested that a friction system consisting of a continuum flowing in a gap and the adjacent local areas of contacting bodies be considered by analogy to a chemical reactor, i.e., as a triboreactor in which exchange processes occur in an open thermodynamic system. Energy and mass transfer and entropy balance equations are derived for such a system. Generalized physical and computational models of the contact friction and wear model are developed on the basis of these equations. It is shown that allowing for the boundary conditions of the problem, the generalized model can be transformed for each specific case of bodies which come into contact. The developed model was experimentally confirmed. Figures 2; references 11.

UDC 621.746.628.047

Examining the Effect of Vibration on a Crystatlizing Melt and the Metal Quality. G. G. Gachechiladze, E. G. Gudushauri, Yu. P. Kirdeyev, G. Ya. Panovko, and V. N. Stroyeva, Moscow, pp 66-70

One possible mechanism by which vibration affects a crystallizing melt is examined. Analytical formulae are cited for determining the conditions under which the growing crystals fracture under the effect of vibration. The results thus obtained are compared to data of model experiments. Certain practical recommendations are given for casting metals under vibration conditions. Figures 6; references 6.

UDC 62-251.755:681.5

Using the Automatic Balancing Criterion to Check to the Serviceability of Automatic Balancing Devices. Yu. G. Zhivotov, Dnepropetrovsk, pp 71-77

A criterion of automatic balancing of rigid rotors with a dynamic instability is proposed. It is shown that this criterion can be used to analyze known and to develop new balancing methods and self-balancing devices, allowing for the rotor dynamics. Examples of analyzing the serviceability of automatic balancing devices with servo systems at subcritical speeds are cited. A dynamic balancing method is developed for rotors spinning at a supercritical speed in order to bring it to the second critical speed; this method is used in the industry. Figures 2; references 7.

UDC 534.014

Unsteady Vibrations of a Rotor-Support System With a Shaking Base. V. F. Shatokhin, Kaluga, pp 78-84

An analysis of the dynamic behavior of the rotor-support system during a pulsed shaking of the foundation is presented. The rotor-support system consists of a spinning rotor of a variable cross section (either ideally balanced or performing forced vibrations due to imbalance) on an oil layer in the bearing and anisotropic elastic mass supports. The seismic pulsed action causes a translational motion of the foundation. The analysis produces a dynamic response by the structure to a given perturbation, including kinematic (displacements, velocities, and accelerations of various points in the structure) and force (bending moments and lateral forces in the rotor cross sections) parameters, as well as dynamic loads on the bearings. Applications of the method and computer programs developed for estimating the serviceability of structures is illustrated by a numerical example. Figures 3; tables 1; references 9.

UDC 621.822.5.039

Determining the Bearing Capacity of a Flat Circular Gasostatic Bearing With a Discrete Pressurization; Pressure Pattern in the Gas Clearance. S. P. Kurpiyanov, Saratov, pp 85-90

A precise mathematical solution of the steady state distribution of the squared pressure in the lubricating layer of a constant thickness in a circular gas-filled bearing with one row of discrete feeders is cited. Based on this solution, the correction factor is found, making it possible to take into account the discrete nature of pressurization almost completely and reduce the search for the bearing capacity of the gas bearing under study by numerical integration to a simple analysis employing the "pressurization line" model. In so doing, the relative calculation error does not exceed tenths of a percentage point. Figures 4; tables 1; references 4.

UDC 539.374

Examining the Straining Mechanism and the Possibility of a Magnetic-Pulse Flanging of Holes. A. M. Baltakhanov, V. B. Yudayev, and S. Ye. Svyatenko, Moscow, pp 91-95

The possibility of making sheet articles by pulsed magnetic field pressure is justified. A two-dimensional numerical model is proposed, enabling us to study the dynamic straining mechanism in a slab allowing for the electromagnetic field diffusion and physical and mechanical properties of the material. The numerical solution was found by the finite elements methods. A numerical experiment where the pulse amplitude is varied makes it possible to control the blank slab behavior during its straining in order to obtain precise articles of a given shape. The high-speed filming of the magnetic-pulse flanging of holes

made it possible to verify the validity of the numerical solution and optimize the straining velocity parameters. Figures 3; references 9.

UDC 681.3.06

Modern Analysis and Random Process Simulation Systems. A. A. Atamanov, A. M. Belov, A. N. Berezkin, V. M. Kostenko, S. K. Kusheverskiy, N. A. Pelykh, S. A. Trofimov, and V. A. Timofeyev, Moscow and Gorkiy, pp 96-101

A number of today's random process analysis and simulation systems is reviewed and their characteristics are compared.

- END -